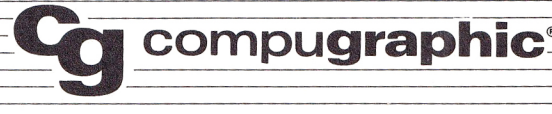
MCS PowerView™ 10 Power Page Option

Training and Applications Manual



MCS PowerView™ 10
Power Page Option
Training and Applications Manual

Prepared by
Educational Services

©Copyright 1984 by
Compugraphic Corporation
Printed in U.S.A.
MCS PowerView is a trademark of Compugraphic Corporation

Part Number 200473-001

TABLE OF CONTENTS

Section	Title	Page
Section I	How To Use This Manual	I-1
Section II	Getting Started	II-1
	Qualifiers	II-2
Section III	Area Composition	III-1
	Page Definition	III-6
	Top and Base Alignment	III-9
	Compartment Definition	III-12
	Linking Compartments	III-14
	Compartment Defaults	III-17
	X/Y Coordinates	III-20
	Vertical Insert Space	III-25
	Vertical Band Set	III-26
	Vertical Band	III-26
	Carding	III-30
	Review Exercise	III-33
Section IV	Formats	IV-1
	Define Format	IV-4
	Check Format	IV-8
	Load Format File	IV-9
	Call Format	IV-12
	Explode Format	IV-14
	Format Variables	IV-19
	Escape Characters	IV-23
	Format Nesting	IV-28
	Passing Variables To Nested Formats	IV-32
	Counters	IV-35
	Qualify Counter Output	IV-41
	Repeat Command	IV-43
	Review Exercise	IV-45

Section V	Pagination	V-1
	Foot Command	V-4
	Head Command	V-5
	Alternating Page Layouts	V-14
	Save Command	V-14
	Restore Command	V-15
	Complex Documents	V-27
	File Management	V-27
Section VI	Applications	VI-1
	Publication	VI-1
	Outline	VI-4
	Schedule	VI-6
	Price List	VI-8
	Display Ad	VI-10
	Ruled Form	VI-12
	Display Ad	VI-14
	Business Cards	VI-16
	Magazine Page	VI-19
	Multiple Page Documents	VI-22
Section VII	Reference	VII-1
	Area Composition Commands	VII-2
	Format Commands	VII-4
	Messages	VII-13
Section VIII	Index of Subjects	VIII-1

How To Use This Manual

Power Page consists of area composition tools and text formatting features. You may use either group independent of the other, or you may combine the two groups for maximum flexibility in job composition and simple pagination.

About This Manual

The Power Page Manual consists of sections that contain discussions and exercises. The discussions are designed to guide you through the various features of the option and to explain how they can best help you perform your work.

This manual is designed to be used at your PowerView 10. As you progress through each section, you will find an exercise for each new command that is discussed. Complete each exercise at your PowerView 10 to be sure that you understand the new material. Consider each new command as a building block; make sure each block is in place before adding another level.

Although step-by-step answers are provided for each exercise, the suggested keystrokes represent *one approach* to the job. It is likely that you will think of other correct solutions to many of the exercises. You should experiment with Power Page to find the best methods for *your* needs.

What's in the Manual?

The Power Page Manual contains eight sections.

Section I, **How To Use This Manual**, is the introduction you are now reading. It provides an explanation of the contents of the manual and an overview of each section.

Section II, **Getting Started**, tells you how to create a User Program Disk (UPD) which includes Power Page and explains how to use qualifiers with Power Page commands. Even if you have already made a set of User Program Disks for your PowerView 10, you should take a few minutes to review the Getting Started Section.

Section III, **Area Composition**, teaches you about new page makeup features. Included in Section III are text and rule positioning commands and Vertical Justification. After studying Section III, complete the review exercise to make sure you understand how area composition tools work together. If you have trouble completing the exercise, go back and study the material; then, try the exercise again. Keystrokes are provided so you can check your work, but remember that there are several ways to approach any typesetting job.

Section IV, **Formats**, steps you through format setup and use. In this section, it is especially important that you follow the sequence of discussions and exercises. A review exercise is included at the end of the section. Use this exercise to gauge your understanding of formats. Again, a solution is provided, but you should experiment with your own ideas and refer to the answers if you need help.

Section V, **Pagination**, teaches you how to combine area composition tools and formats to produce multiple page documents. Included in the Pagination Section are header and footer setups for simple and complex documents and new file management commands and techniques to help you use Power Page more efficiently.

Pagination requires a combination of area composition and formatting commands. *Before you begin Section V, you should complete Sections III and IV.*

Section VI, **Applications**, contains a variety of sample jobs to give you more practice with specific applications. Each application consists of a typeset job sample followed by an explanation and illustrations of the format file and job file. You may not wish to complete all the application exercises. However, pay special attention to the applications that apply to typical jobs you do on a regular basis. Keep in mind that the explanations following the applications do not represent the only way to complete a job. Try to finish the application without looking at the answers.

Section VII is a **Reference Guide** designed to give you quick access to specific commands and procedures. Contained in the Reference Guide are hardware and software specifications, all format and area composition commands, and error and status messages.

The Reference Section also summarizes discussions and procedures that are outlined in the manual. A detachable quick reference card in the back of the manual lists and briefly explains each Power Page command. You may want to remove the card and keep it near your PowerView 10.

Section VIII is an **Index**. Arranged alphabetically by subject, it is located at the end of the manual and is designed to help you quickly locate information.

☛ **NOTE:** *This manual assumes a working understanding of input, editing, file management and system setup. Refer to your* **PowerView 10 Reference Manuals** *and the* **PowerView 10 Fast Start Manual** *if you need to refresh your memory.*

GETTING STARTED

User Program Setup

Whenever you receive a new option, you must create a User Program to include that option. The following steps will help you create a User Program which includes Power Page. See the **PowerView 10 Fast Start Manual**if you need additional help.

1. Format three blank disks as double-sided program disks.
   - ► Insert the Disk Utility into disk drive 1 (F1).
   - ► Press the Load button on the front of the PowerView 10 controller.
   - ► When the menu appears on your screen, enter **1** for the Format Disk Menu, fill in the time stamp and press Execute.
   - ► When the second menu appears, enter the volume name, enter **3** for Dual Sided-Program Disk and press Execute.
   - ► When all three disks have been formated, lable them Volume 1, Volume 2, and Volume 3, and continue with step 2.
2. Create a User Program.
   - ► Insert the System Generator Disk into disk drive 1 (F1).
   - ► Press the Load button on the front of the PowerView 10 controller.
   - ► Answer the questions that appear on the screen. Your answers depend on the configuration of your system and the options you have. The questions that pertain to Power Page are:

     ??Is the background compose option selected for the target system
     ??(Y or N; default N)?
     Select **Y** (Yes).

     Is the paging option selected for the target system
     (Y or N; default N)?
     Select **Y** (Yes)

3. After answering the questions, follow the instructions on the screen to transfer programs from Master Program Disks (MPDs) to the User Program Disks.

Using Qualifiers in Commands

Although you usually define commands in picas and points, you may use inches, millimeters or ciceros in commands. To instruct the PowerView 10 to use an alternate counting unit, a qualifier must be entered into the command. The available qualifiers are:

Counting Unit	**Qualifier**
Metric (millimeter)	M
Cicero	C
Inch	I
Picas and Points	P

To access an alternate counting unit, enter the qualifier after the number in the command. For example:

Line Length of 8 inches	<LL8I>
Line Length of 75 millimeters	<LL75M>
Advance Lead 2 inches	<ALD2I>
Linespace of 1 cicero	<LS1C>

Commands containing several number fields, such as tabs or cuts, require a qualifier after each number. All qualifiers may be used in any power page command. For more details on using qualifiers, refer to the **PowerView 10 Reference Manuals.**

Now you are ready to use the Power Page option. Turn to Section III to learn about area composition tools.

AREA COMPOSITION

Area composition tools on the PowerView 10 will help you to do area composition and full page makeup without cutting and pasting the output.

This section of the manual describes the procedures that simplify area composition on a single page. These procedures will also apply when you begin working with multiple page documents.

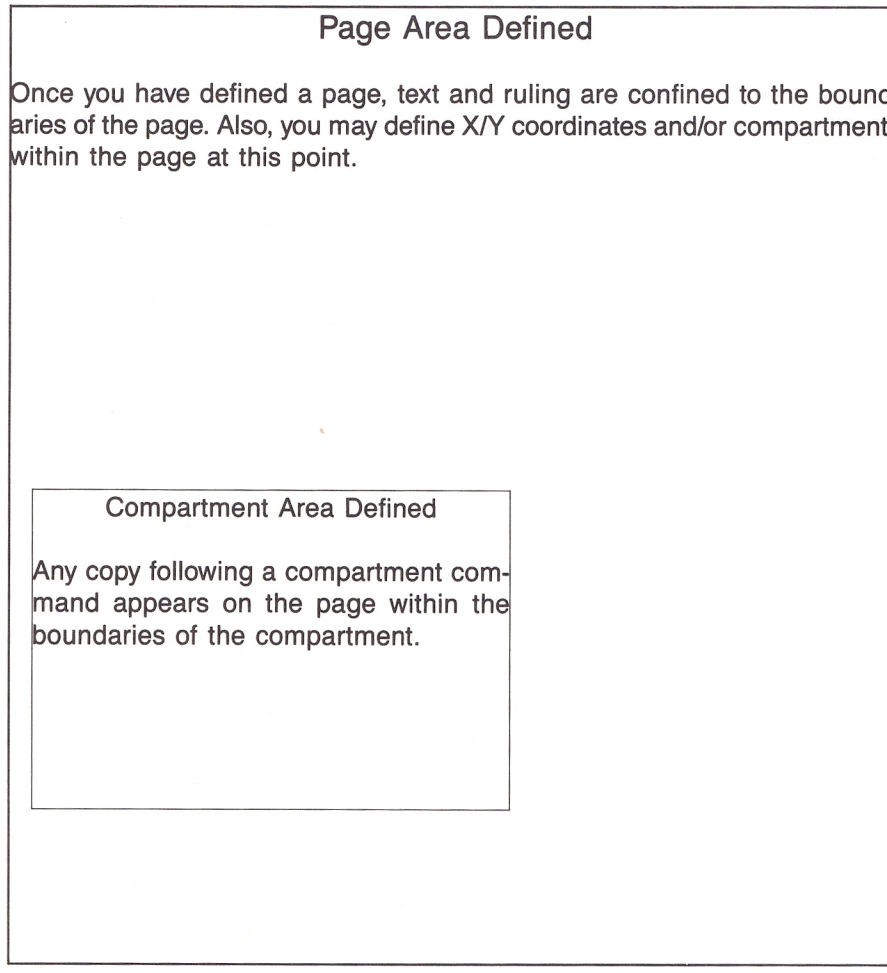
Overview of Area Composition Commands

Using the *page* command, you may define the space in which you wish to work. If you exceed the boundaries of the defined area, you will see an error message on the screen.

Page Area Defined

Once you have defined a page, text and ruling are confined to the boundaries of the page. Also, you may define X/Y coordinates and/or compartments within the page at this point.

After entering the page command, you may define *compartments* within the page. A compartment is a defined area within the page where text and rules are contained. You may define as many compartments as you like as long as they fit within the defined page area. When text has filled the first compartment, additional text can flow into the second defined compartment.



Defining a *page* allows you to use *X/Y coordinates* to position text or rules on the page. Any text or rules following an XY command are repositioned on the page. Once a compartment is defined, the XY command can only position items within the compartment. XY commands can be used repeatedly to position as many items as you like.

Page Area Defined

Once you have defined a page, text and ruling are confined to the boundaries of the page. Also, you may define X/Y coordinates and/or compartments within the page at this point.

- When an X/Y coordinate is defined, the copy is positioned on the page at the position specified in the command.

Compartment Area Defined

Any copy following a compartment command appears on the page within the boundaries of the compartment.

- When an X/Y coordinate is defined within a compartment, copy is positioned in the compartment relative to the borders of that compartment.

Vertical justification commands allow you to distribute excess vertical space automatically within a compartment. Excess vertical space may be placed above or below the text or between each line of copy to fill the compartment.

Page Area Defined

Once you have defined a page, text and ruling are confined to the boundaries of the page. Also, you may define X/Y coordinates and/or compartments within the page at this point.

When an X/Y coordinate is defined, the copy is positioned on the page at the position specified in the command.

Any copy following a compartment command appears on the page within the boundaries of the compartment.

With vertical justification, excess vertical space can be distributed between each line of text to fill the compartment.

Area Composition Commands

Page Command<PAGEw,d,g>

The *page* command *must* be the very first entry in your job. This command will put your PowerView 10 into the page mode. Page mode allows the PowerView 10 to recognize other commands associated with the paging option. The default mode of your system is galley mode. Once the page is defined, you may use compartments, X/Y coordinate commands, vertical justification or ruling commands, but *you may not use multiple column* commands.

When you enter Page mode, the letter "P" will appear in the far left corner of the third status line. When you are in galley mode—no page defined—the letter "G" will appear in the far left corner of the third status line.

The following is an example of a *page* command:

<PAGE45,40,6>

The *width* of the page is the first number. In this example the page is 45 picas wide. You can enter the command with picas and points as well. If the job was 40 picas and 6 points wide, the number entered would be *40.6*. The *default* width value is the typesetter's maximum vertical range.

The *depth* of the page is the second number. In this example, the page is 40 picas deep. You can enter this command in picas and points. The *default* depth value is the typesetter's maximum vertical range.

The last number in the page command is the *interpage gap*. In multiple page jobs, this is the amount of advance leading you want added between output pages. The *default* gap value is zero.

The maximum width, depth and gap values allowed in the page command depend on the typesetter's maximum horizontal and vertical ranges. Once the

page command is entered, your line length automatically becomes the width of the page, and all text or ruling must fall within the defined boundaries. The page command must be the *first entry in the file*. If the page command is not at the beginning of a file, a "Galley Mode" error message appears.

You may use multiple page commands in a file to define new dimensions for output pages. Each page command must be the *first entry on both the file page and the new output page*. The message "Page Open" means that the page command is not the first entry on the file page.

The space defined in the page command does not have to be the size of a printed page. You may define the page command to match the dimensions of your printed page and have copy fill the space. However, if you wish to include crop marks or margins, your page command must define a space larger than the printed page. You must then position copy within the page.

A Note About Exercises

Now you are ready to complete an exercise using the area composition command just explained. All exercises in this manual were set up and typeset using the Triumvirate Regular, Italic and Bold typefaces. Your typeset output may look different than the examples you see here, depending on the typeface you use.

The exercises assume you are using disk drive one and do not specify a disk drive. If an error message appears while completing an exercise, turn to the Reference Section (pages VII-20 through VII-23) in the back of the manual for an explanation.

Exercise 1—Defining a Page

The following exercise will teach you to enter a *page* command.

You Are Going To:

1. Start a new file called "area".
2. Define a page 28 picas wide and 30 picas deep with an interpage gap of 3 picas.
3. Enter parameters for text.
4. Enter text in the page.

Do The Following:

► Open a new file called "area":

NEW FILE area EXECUTE

► Define the page dimensions:

TYPE CMO PAGE28,30,3 TYPE CMX

► Enter <PS10<LS12>.

► Enter the following text:

Once you have defined a page, all text and ruling will be confined to the boundaries you define. There is no reason to define a line length because the page width determines the line length for you.

► LEFT RETURN

►Examine the view window to see how the page command affects the copy.

►Append the file:

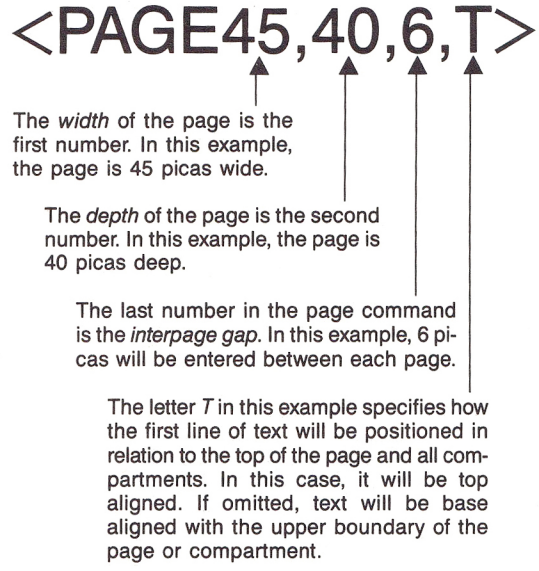
APND EXECUTE

►Close the file:

CLOSE FILE EXECUTE

Page Command (Cont.)—Top and Base Alignment

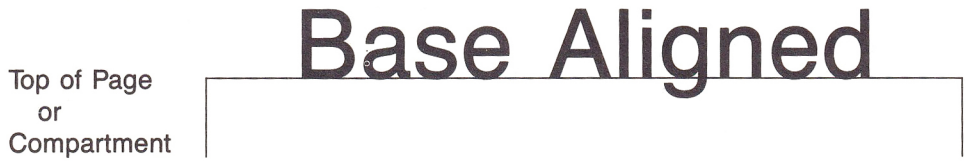
The page command can include an alignment argument to allow text to be *top* or *base aligned* with the *top* of a page and all compartments on a page. If the page command specifies top alignment, all text in compartments in that file will be top aligned. If you do not specify top alignment, the page command defaults to base alignment.



When the *T* is added to the page command, copy will *top align* with the page and, if there are any compartments in the page, all copy in the compartments will be top aligned.

The following diagrams show you how top and base align features of the page command work.

<PAGEw,d,g>
If you do not enter a T in the *page* command (leave it blank), the copy in the page will be *base aligned* with the *top* boundary of the page. For example, if the first line of text is 36 point, the first line will appear above the top of the page. The following diagram represents this:



Text within compartments in this page will be base aligned with the top of the compartment.

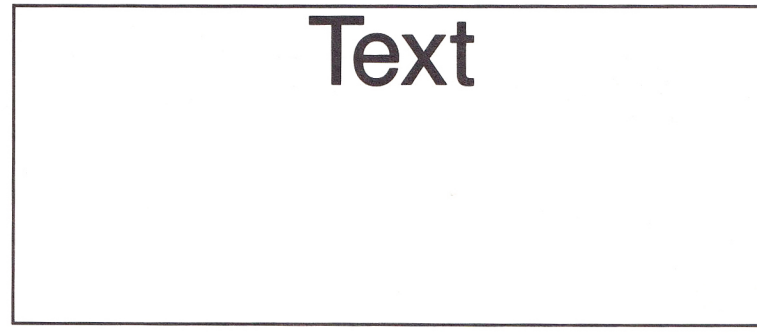
<PAGEw,d,g,T>
If you enter a page command and use the T to specify top alignment, the first line of text will be top aligned with the top of the page. The following diagram illustrates how this affects your output copy:

In this example, the top of the first line of text top aligns with the upper boundary of the page. If compartments are used within this page, the copy in the compartments will top align with the top of the compartments.

When text is top aligned, the PowerView 10 will lower the baseline so that text will be placed within the compartment. Because of this, working with rules in top aligned compartments may be tricky. If you wish to set a box first and then set text in the box, you must "override" the top alignment by positioning the box with an X/Y coordinate (generally, XY0,0). The XY command turns off the top alignment for that compartment, so you must advance lead to set the text once the box has been entered; or you may redefine the compartment once the rules have been set.

The diagram below shows how output of this example might look and also the commands to do it.

Top of Page



The keystrokes used:

<PAGE24,10,5,T><XY0,0><DBX100,120><ALD30><PS36>Text↔

OR

<PAGE24,10,5,T><XY0,0><DBX100,120><CMPT0,.6,24,10,J,U><PS36>Text↔

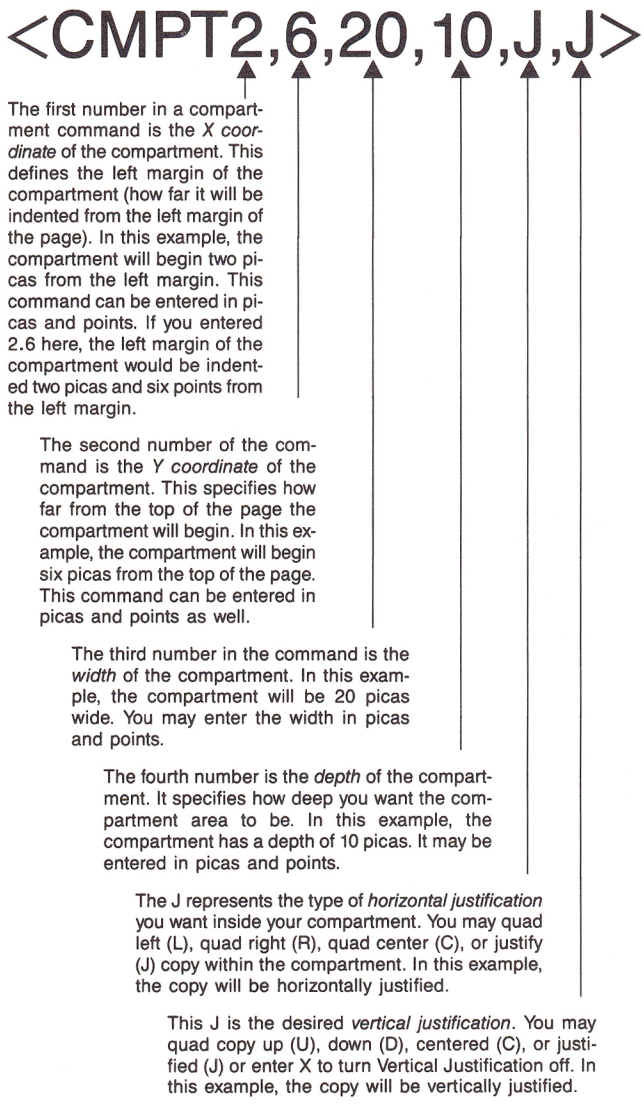
In the second set of keystrokes, the .6 depth value in the compartment command lowers the baseline so "Text" does not touch the top of the box.

Using default compartment values in the second set of commands will save keystrokes. With defaults, the commands to set the diagram above look like:

<PAGE,24,10,5,T><XY0,0><DBX100,120><CMPT,.6,,,J,U><PS36>Text↔

Compartment Command<CMPTx,y,w,d,h,v>

The compartment command creates a *compartment* or specified area for text placement within a page. You may not use multiple column commands inside a compartment. Cut-runarounds may be used only if Vertical Justification is turned off. The following is an example of a *compartment* command:



☛ **NOTE:** *Vertical Justification will be discussed later in this section.*

The compartment boundaries *cannot exceed the page command boundaries.* Exceeding the size of the page will result in the message "Page Exceeded". Once a compartment is defined, all text or rules following it will be placed within the compartment. Any text or rule positioning commands used within the compartment cannot exceed the compartment boundaries.

After you define a compartment, you must work within compartments for the rest of the page. The only way to work in other areas on the page is to define another compartment. To work within the whole job area again, define a compartment the same size as the page.

Exercise 2—Defining a Compartment

In the following exercise, you will position copy on a page using a compartment. The location for the compartment is specified in the exercise.

You Are Going To:

1. Call the file "area" to the screen.
2. Define a compartment.
3. Enter text in the compartment.

Do The Following:

► Call the file "area":
[CALL PAGE] area [EXECUTE]

► Move the cursor to the end of the job:
[LIMIT] [SCRL UP]

► Define the compartment:
[TYPE CMO] CMPT2,6,11,9,J,U [TYPE CMX]

► Enter the following text:

Any text or ruling following a compartment command will appear on the page in the area defined by the compartment.

► LEFT RETURN

The view window of your PowerView 10 should look like this:

Once you have defined a page, all text and ruling will be confined to the boundaries you define. There is no reason to define a line length because the page width determines the line length for you.

Any text or ruling following a compartment command will appear on the page in the area defined by the compartment.

► REPL PAGE EXECUTE

► CLOSE FILE EXECUTE

Linking Compartments

Many jobs have text and ruling placed in more than one area of the page. By defining several compartments in a row, text will automatically flow from one compartment to another. This is called *linking* compartments.

A maximum of ten compartments may be linked together at one time. More than one set of linked compartments may be defined per page as long as text or rules are entered to separate them.

Exercise 3—Using Linked Compartments

The following exercise will show you how to use linked compartments in a page. When the copy has filled the first compartment, you will notice a <?FB> command appears where the copy "breaks" from one compartment to the next. This command is automatically inserted to let you know where the compartment ends, but does not require you to do anything.

You Are Going To:

1. Call the file "area" to the screen.
2. Add a new compartment command to the job.
3. Enter more text in the job.

Do The Following:

- ► Call the file "area" to the screen.
- ► Position the cursor immediately *after* the first compartment command.
- ► Enter the second compartment command, which should look like <CMPT15,5.6,11,9,J,U>.
- ► LIMIT SCRL UP
- ► Enter the following text:

  For copy to flow from one compartment to the next automatically, the compartment commands must be linked. When the text has filled the first compartment, it will automatically "jump" into the next compartment. You will know where the copy ended in the first compartment when the <?FB> command appears. If your view and source windows are attached, you will see the text flowing flowing from one compartment to the next.

- ► LEFT RETURN

The view window of your PowerView 10 will look like this:

Once you have defined a page, all text and ruling will be confined to the boundaries you define. There is no reason to define a line length because the page width determines the line length for you.

Any text or ruling following a compartment command will appear on the page in the area defined by the compartment. For copy to flow from one compartment to the next automatically, the compartment commands must be linked. When the text has filled the first compartment, it will automatically "jump" into the next compartment. You will know where the copy ended in the first compartment when the (?FB) command appears.

The source window of your PowerView 10 will look like this:

<PAGE28,30,3><PS10<LS12>Once you have defined a page, all text and ruling will be confined to the boundaries you define. There is no reason to define a line length because the page width determines the line length for you.←
<CMPT2,5.6,11,9,J,U><CMPT15,5.6,11,9,J,U>Any text or ruling following a compartment command will appear on the page in the area defined by the compartment.←
For copy to flow from one compartment to the next automatically, the compartment commands must be linked. When the text
<?FB> has filled the first compartment, it will automatically "jump" into the next compartment. You will know where the copy ended in the first compartment when the "(?FB)" command appears.←

S:D:Fi N:area Pg:01
PS10.0 LS 12.0 FT 1 L 00:04 A TB XX D:10.06 I
P SF:100 ATT

► REPL PAGE EXECUTE
► CLOSE FILE EXECUTE

Compartment Defaults

Using default values in compartment commands will save keystrokes during text input and editing. If values are not entered in the fields of compartments, the command arguments will default as follows:

Argument	Default
X (horizontal indent)	Last entered X value, or 0
Y (vertical depth)	With top alignment: last entered Y value, plus compartment depth, plus a linespace, minus the top alignment value With base alignment: last entered Y value, plus compartment depth, plus a linespace; or 0
W (compartment width)	Last entered width, or width of page minus X value
D (compartment depth)	Last entered depth value, or depth of page minus Y value
H (horizontal justification)	J (Justify) is default
V (vertical justification)	J (Justify) is default; if no VIS or VB commands entered, copy will quad up

When using default values, there may be instances when the X, Y, W or D arguments of the compartment command will display as DATA symbols in the display parameters. This will occur when linking compartments and before the next linked compartment is activated. At that point, the DATA symbols will be replaced with the appropriate values for that compartment.

Force Break Command

You may wish to go from one compartment to the next without waiting for the automatic break. To do this, enter the command <FB> (force break) where you want the copy to change compartments. The copy following the <FB> command will be in the next compartment.

A force break is used to move copy from the current compartment to the next before the current compartment depth is reached. When the final compartment is reached, the only way to position text elsewhere on the page is to define a new compartment.

Exercise 4—Using A Force Break in Linked Compartments

This exercise will show you how to change from one linked compartment to the next without waiting for the PowerView 10 to make that decision. The exercise tells you where to enter the force break command.

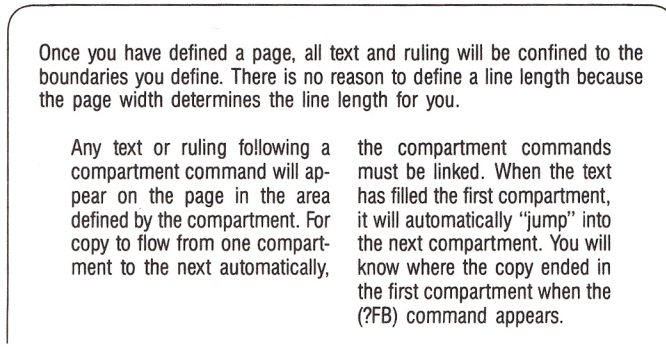
You Are Going To:

1. Call the file "area" to the screen.
2. Enter a force break command in the text.
3. See how the force break affects the copy displayed in the view window.

Do The Following:

- ► Call the file "area" to the screen.
- ► Position your cursor at the comma after the phrase "*to the next automatically,*"
- ► [LEFT] [RETURN]
- ► Enter a force break command:
  [TYPE CMO] FB [TYPE CMX]

The view window of your PowerView 10 should look like this:



► REPL PAGE EXECUTE
► CLOSE FILE EXECUTE

An output page equals one file page. When the last compartment on a page is filled, the "Page Overset" message appears. This is an automatic warning instructing you to append or replace the page. You will not be able to enter more text until the page on the screen is stored on the disk.

Possible Error Messages:

PAGE EXCEEDED
The compartment you have defined exceeds one or more of the page boundaries.

CMPT LIMIT
You have linked more than ten compartments.

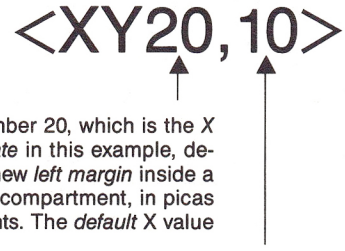
NO COMPT
You have entered a force break when no compartment is defined.

PAGE OVERSET
You have more copy than will fit on the page.

X/Y Command <XYi,d>

The X/Y command defines a specific point in a page or compartment where the text/ruling will begin. The coordinates in an X/Y command are relative to the top left corner of the page. Once a compartment is defined, the X/Y command values will be relative to the top left corner of the compartment.

For instance, to position copy 20 picas from the left margin of a page and 10 picas from the top, the X/Y command would look like this:



The number 20, which is the *X coordinate* in this example, defines a new *left margin* inside a page or compartment, in picas and points. The *default* X value is zero.

The number 10, in this example, is the *Y coordinate*. It defines the *vertical position* of the copy or ruling, in picas and points. The *default* Y value is zero.

Once the X/Y command is entered, all text and commands (except compartment and X/Y commands) are referenced from that point. For instance, any tabs you may have set will use the X coordinate in the X/Y command as the new left margin. Therefore, if your "X" is 2 and a tab is set to start at an indent of 0, the tab will start at the indent of 2 picas specified in the X/Y command. To return to the left margin of the page, a new X/Y command must be set.

Each time an X/Y command is used, the line length is automatically reset. The new line length is the space between the indent and the right margin of the page or compartment. For instance, if your page width is 40 picas and you set an "X" coordinate of 12, your line length will automatically be reset to 28 picas.

Exercise 5—X/Y Positioning

The following exercise will show you how to use an X/Y command to position copy in a page.

You Are Going To:

1. Call the file "area" to the screen.
2. Position text in the page with X/Y coordinates.

Do The Following:

- ► Call the file "area" to the screen.
- ► Position the cursor at the beginning of the first paragraph.
- ► Enter the X/Y coordinate command:
  [TYPE CMO] XY5,17 [TYPE CMX]

The view window of your PowerView 10 should look like this:

Any text or ruling following a compartment command will appear on the page in the area defined by the compartment. For copy to flow from one compartment to the next automatically, the compartment commands must be linked. When the text has filled the first compartment, it will automatically "jump" into the next compartment. You will know where the copy ended in the first compartment when the (?FB) command appears.

Once you have defined a page, all text and ruling will be confined to the boundaries you define. There is no reason to define a line length because the page width determines the line length for you.

- ► [REPL PAGE] [EXECUTE]
- ► [CLOSE FILE] [EXECUTE]

Notice how the XY command moved the first paragraph. The other two paragraphs were not moved because their position is defined by compartments. Try typing a different set of numbers in the XY command and see what happens.

Possible Error Messages:

CMPT EXCEEDED
A specified X/Y position is outside the compartment boundary.

PAGE EXCEEDED
A specified X/Y position is outside the page boundary.

Horizontal and Vertical Justification—A Comparison

To understand vertical justification, it is helpful to compare it to horizontal justification. The following discussion compares the two.

Horizontal justification allows the PowerView 10 to typeset copy with even left and right margins. Each typeset line of copy "spreads" to fill the line length. To do this, the PowerView 10 can divide excess space among the spaces between words (spacebands) and the space between letters (letterspacing).

The amount of extra space entered between words and letters can be defined by you. Entering spaceband and letterspacing values gives you control of how much excess horizontal space will be entered. To override excess space within a line, text can be quadded left, right or centered.

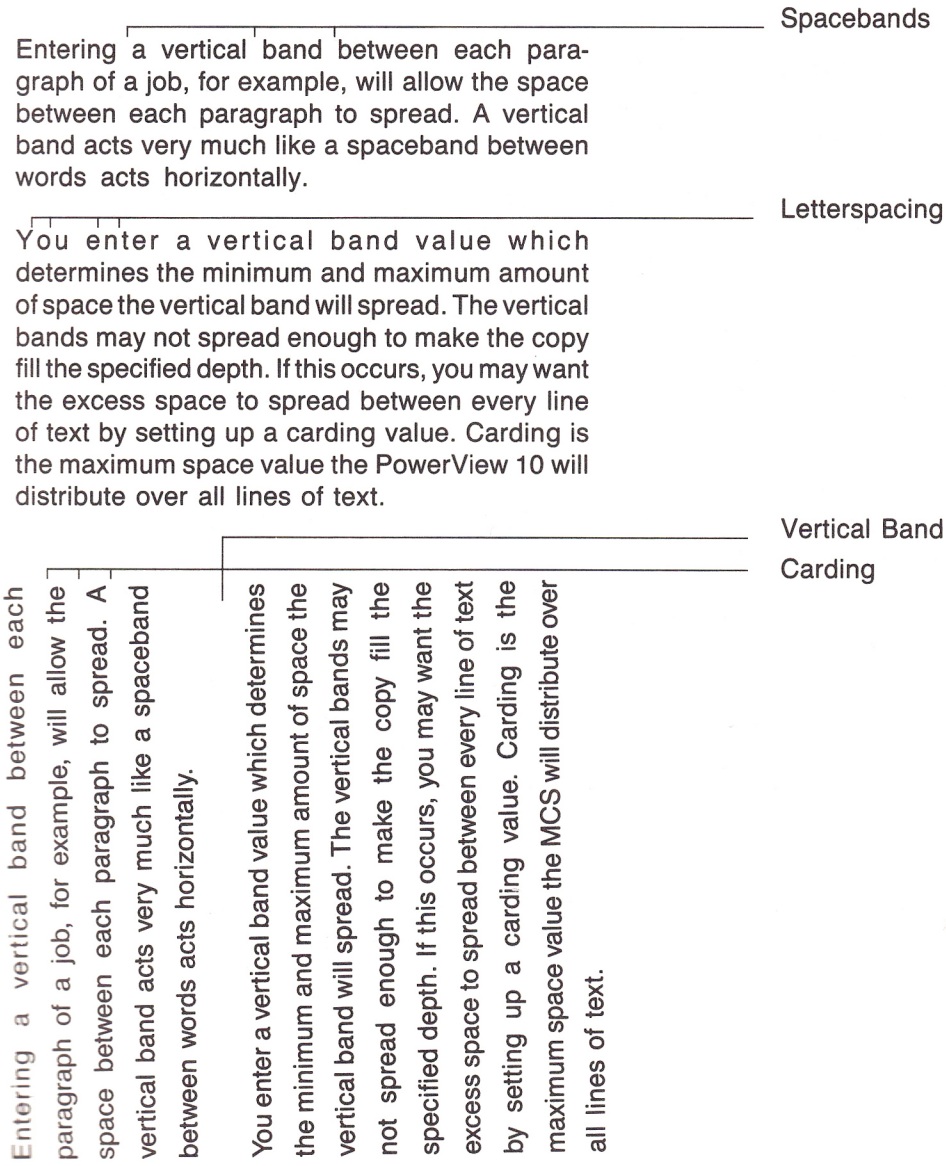
Vertical Justification allows the PowerView 10 to add extra space between lines of text to fill a selected depth. To do this, it uses vertical bands and carding values.

Entering a *vertical band* between each paragraph of a job can be compared to spacebands between words. They will allow the space between each paragraph to spread so the copy will fill the entire depth. The vertical band value (which determines the minimum and maximum amount of space the vertical band will spread) can be defined by you.

If the vertical bands do not spread enough to make the copy fill the depth, you may want the excess space to divide among every line of text. This can be done by setting a *carding* value. Carding can be compared to letterspacing. It is the automatic addition of space between every line of text. A maximum value can be specified which will be divided equally among all lines in a compartment.

Just as spacebands and letterspacing combine to justify a line of text horizontally, vertical bands and carding combine to justify text vertically.

The following diagram compares horizontal and vertical justification:



Vertical Justification

Vertical Justification On/Off

Vertical justification is included with Power Page. With this feature, the PowerView 10 automatically distributes excess vertical space as specified. Vertical justification may be defined in a compartment or outside of a compartment using the Vertical Justification On (VJO) and Vertical Justification Off (VJX) commands. The VJO and VJX commands must be on the same file page or the PowerView 10 will give you an error message. Refer to your **PowerView 10 Reference Manuals** for more information on VJO and VJX commands.

Vertical justification is turned on automatically by the compartment command. Because vertical justification is already on, **<VJOddd,q>** command is illegal. To use this command, vertical justification must be turned off (X) in the compartment command.

When defining the compartment definition, the parameter for vertical quadding may be entered as J (justified), U (up), D (down), C (centered) or X (Vertical Justification Off).

Vertical Insert Space (VIS)

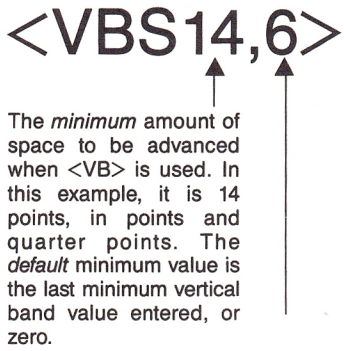
The VIS command defines the point where excess vertical space is inserted during justification. When vertical justification is turned on within a compartment, the VIS command overrides any vertical quad command.

You may also use the VIS command to sepcify a proportional amount of space. The command sequence is <VISx>. The *x* represents a proportional value. For example, a <VIS2> command will expand twice as much as a <VIS> or a <VIS1>.

Vertical Bands

A vertical band functions like an Advance Lead command that can expand automatically to fill a compartment or VJO depth. You define the minimum value plus the maximum expansion value.

The vertical band set command looks like this:



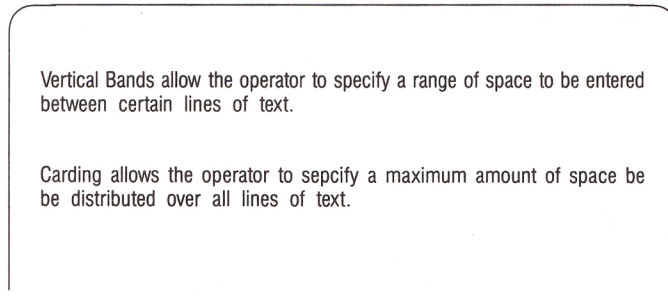
To activate this value, the <VB> command is entered in the text. The vertical band set command is used to set a value; it does not take effect until you enter a <VB> command.

When a vertical band is entered in a job, the minimum space is in effect. If excess space remains at the end of a compartment, it is divided among all vertical bands until the maximum expansion is reached. Vertical bands will only expand to maximum when Vertical Justification is Justified. Quadding Up, Down or Center forces all vertical bands to the minimum value.

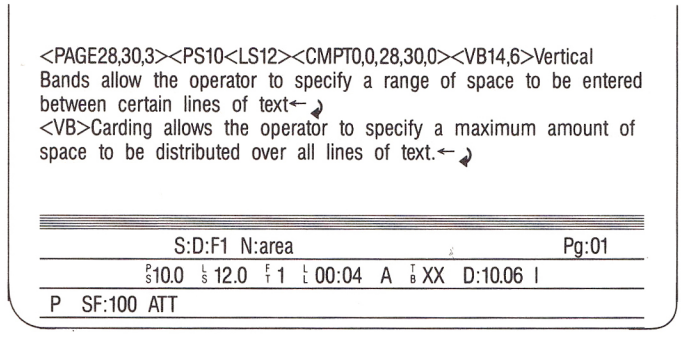
If a vertical insert space is used, all vertical bands are set to minimum, and any excess space is divided among all vertical insert spaces.

Suppose you have a text file to typeset. The customer specifies that there can be 14 to 20 points of *extra* space between each paragraph, in addition to the linespace. The page size of the job is 28 picas wide and 30 picas deep. The compartment size is the same as the page size with centered horizontal quadding and justified vertical quadding.

The view window of your PowerView 10 will look like this:



The source window of your PowerView 10 will look like this:



Exercise 6—Using Vertical Bands

You Are Going To:

1. Start a new file called "vertical."
2. Define a page.
3. Define a compartment.
4. Enter a vertical band set command.
5. Enter text using vertical bands to add space between paragraphs.

Do The Following:

► Open a new file called "vertical":
[NEW FILE] vertical [EXECUTE]

► Define page dimensions:
[TYPE CMO] PAGE28,30,0 [TYPE CMX]

► Define compartment dimensions using page width and depth defaults and horizontal and vertical justification defaults:
[TYPE CMO] CMPT,,,,, [TYPE CMX]

► Define a vertical band set command:
[TYPE CMO] VBS12,6 [TYPE CMX]

► Enter <PS10<LS12>

► Enter the following text using quad left, return to end the paragraphs:

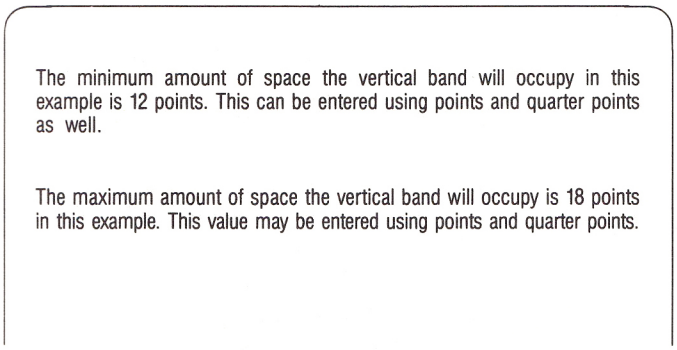
The minimum amount of space the vertical band will occupy in this example is 12 points. This can be entered using points and quarter points as well. [LEFT] [RETURN]

<VB>The maximum amount of space the vertical band will occupy is 18 points in this example. This value may be entered using points and quarter points. [LEFT] [RETURN]

► [APND] [EXECUTE]

► [CLOSE FILE] [EXECUTE]

The view window of your PowerView 10 will look like this:



The space between paragraphs in the output should measure from 24 to 30 points. Remember, a vertical band is added to the active linespace. Since the linespace is 12 and the minimum vertical band is 12, the minimum space between paragraphs will be 24 points and can expand to 30 points.

CARDING <CRDdddd.d>

The carding command in a job might look like this:

The 14 represents the *maximum* amount of space, in points, that the PowerView 10 will divide among all lines of text. The *default* carding value is zero.

A carding value represents the maximum amount of space that will be divided among all lines to reach the desired depth within the page or compartment. When a carding command is entered, the PowerView 10 will spread space between every line of text that ends with a leading advance command (return, tab return). A carding value is added to the current linespace value.

The maximum legal carding value is the maximum vertical range of the typesetter. Carding values may be entered in points and quarter points.

When used with vertical bands, carding will not occur until the maximum vertical band is in effect. Carding will work only with a justified vertical quad. Vertical insert space, quad up, quad down or quad center will turn off carding.

You can control the location of carding by the position of the carding command. To turn carding off, set a carding value of zero.

Exercise 7 — Carding

The customer in Exercise 6 (Vertical Band exercise) required the following:
Page Size 28x30
Point Size 10
Linespace 12
Compartment size 0,0,28,30,J,J
Vertical bands with a size range of 12 to 18 points.

You can use carding to spread the copy to fill the page depth.

Exercise 7—Carding

You Are Going To:

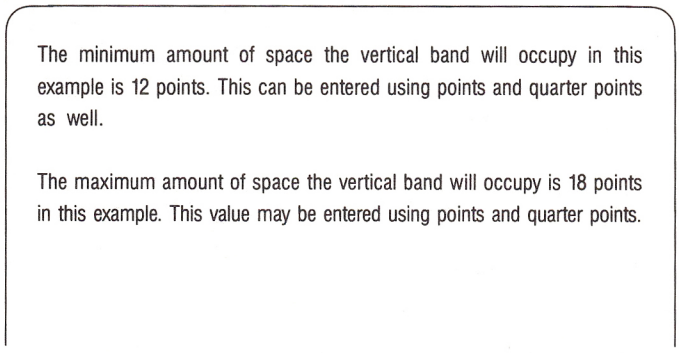
1. Call the file "vertical" to the screen.
2. Enter a carding command.

Do The Following:

- ► Call the file "vertical" to the screen.
- ► Position the cursor on the first character of text.
- ► Enter the carding command to allow up to 14 points of space to be divided among all lines.

  TYPE CMO CRD14 TYPE CMX

The view window of your PowerView 10 will look like this:



► REPL PAGE EXECUTE
► CLOSE FILE EXECUTE

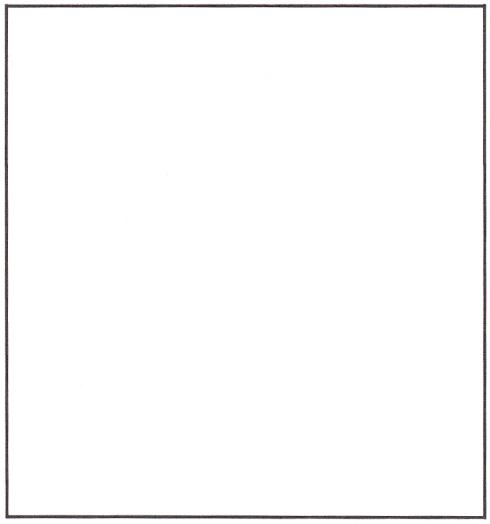
Notice that the space between lines of text in the output is larger than the linespace. The extra space was added by the PowerView 10 to justify the text vertically. There could be up to 14 additional points of space divided among all lines of text based on the carding value you entered.

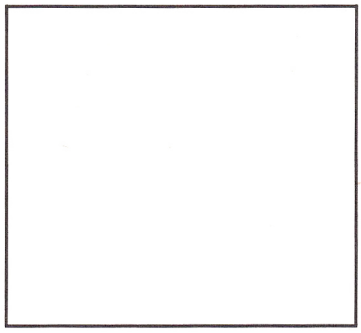
Area Composition Review Exercise

The review exercise on the following pages will show you how the area composition features can benefit page layout work. Complete the exercise, referring to the screen representation if you want to see how the job will look in your text file.

Cross-Country Gear

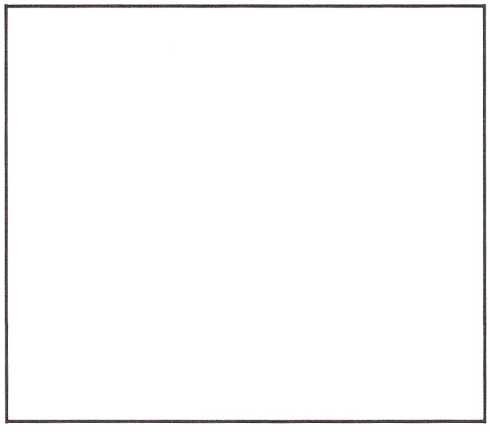
For sport and for safety. . . **Camera Corporation's** compact 8—20 ABC model binoculars, $130; 8700 Compass, $199.45; **Photo Corporation's** XZ-2 pocket camera, $89.95; **Moonbeam Creation's** original "Snuggler Strap" for securing that camera while you ski (not pictured), $23.95 for nylon, $38.95 for leather.





For serious mountaineers skiing serious mountains. . . **Live-Link** offers Shovel III, $28.05; Slope Meter, $11.55; and Avalanche Ski Pole Probe, $55. **Salawe** adjustable crampons available through **Johns Mountaingear**, $28.35.

Slip-Sliding away. . .at least with luck, some snow, and any one of these cross-country ski waxes. Extensive waxing kits from **Swish** and **Koko** (prices vary) are excellent when packed in your truck for quick access. For the part of your ski where wax will not adhere, try a variety of Hollow sprays. Prices available upon request.



The completed job will look like this in the source window:

<PAGE36,53,6><PS12><KO>Cross-Country Gear←↵
<XY15,1><DBX100,192,0,15><XY0,19><DBX100,120,,11>
<XY15,30><DBX100,156,,15>↵
<CMPT0,1,14,16,L,C><CMPT13,19,17,12,L,C><CMPT0,32,14,13,L,C>↵
<PS10<LS11>For sport and for safety. . .↵
<FT3>Camera Corporation's<FT1> compact 8—20 ABC model binoculars, $130; 8700 Compass, $199.45; <FT3>Photo Corporation's<FT1> XZ-2 pocket camera, $89.95; <FT3>Moonbeam Creation's<FT1> original "Snuggler Strap" for securing that camera while you ski (not pictured), $23.95 for nylon, $38.95 for leather.↵
<FB>For serious mountaineers↵
skiing serious mountains. . . <FT3>Live-Link<FT1> offers Shovel III, $28.05; Slope Meter, $11.55; and Avalanche Ski Pole Probe, $55.
<FT3>Salawe<FT1> adjustable crampons available through <FT3>Johns Mountaingear,<FT1> $28.35.↵
<FB>Slip-Sliding away. . .at least with luck, some snow, and any one of these cross-country ski waxes. Extensive waxing kits from <FT3>Swish<FT1> and <FT3>Koko<FT1> (prices vary) are excellent when packed in your truck for quick access. For the part of your ski where wax will not adhere, try a variety of Hollow sprays. Prices available upon request.↵

If you choose to enter the boxes after entering a copy block, you must use compartments to position the rules.

After completing the Area Composition review exercise, begin the next section on Formats.

FORMATS

What Is A Format?

A format is a group of characters which can be used over and over again with a minimum of keystrokes. They usually describe the way something looks. They are assigned a name and stored in a format file. To use them in a job, you type the name you assigned the group of characters where you want them to appear. The format name makes the characters "active" in your job, but you won't see them on the screen; only the format name will be visible.

Why Is It Used?

Formats are used to simplify input. Not only do they reduce the number of keystrokes used to type a job, they reduce the number of characters visible on your screen. This makes editing and proofing much easier. They also allow you to fit more of your job on the screen at one time. Long jobs will use less disk space and less screen space.

Formats are also used to simplify editing. Any change can be made once in a format file, and the change is automatically made throughout the entire job wherever the format is used.

In addition, formats make it easier for inexperienced operators to type complex jobs. An experienced operator can create a set of formats that anyone can use.

How Does A Format Differ From A UDK?

Formats and UDKs really perform the same function, but with one *major* difference. Formats, like UDKs, can be used over and over once they are defined. However, each time a UDK is used, all the keystrokes it contains are "burst" onto the screen. When a format is used, you see only the format name, not the codes it contains.

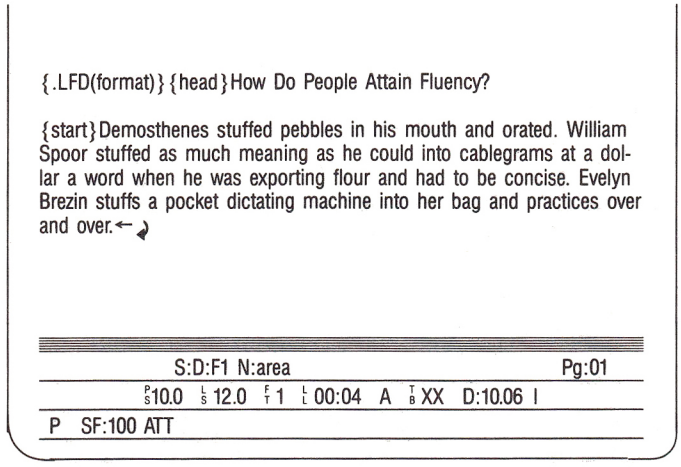
Formats and UDKs can be used together in a job to automate your production flow. UDKs are best used for small strings of repetitive characters, and formats for longer strings.

How Are Formats Created And Used?

When using formats, you work with two files. One file, "the format file," contains all the format definitions and the names you have given them. This file is created before you work on your job.

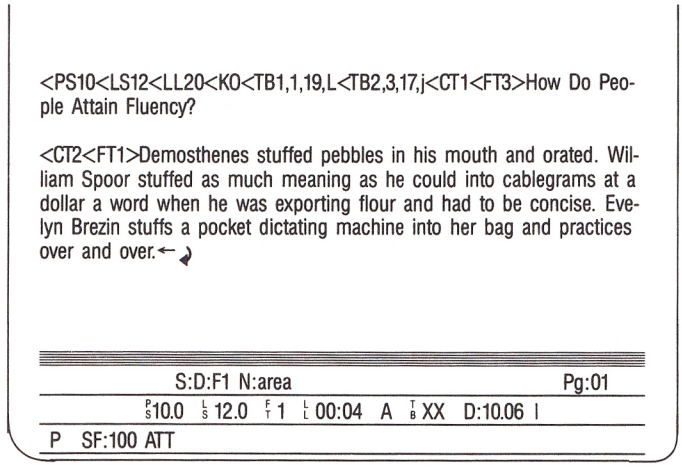
The second file, your "job file," contains the job you are setting. The job is set as you would set any other, but two things are different. First, you must indicate the name of the format file. Second, you will enter the format names as you need them.

If you are using formats, the source window of your PowerView 10 might look like this:



Text with Format Calls

This is how the same job might look without formats:



Text with Parameters

Format Files

Format files are stored as text files on disk. The only difference between a format file and a regular text file is the format file contains format definitions. All the functions used with an ordinary text file can also be used with a format file except that *the Edit mode must be used to enter or edit any format definitions*. The error message "Fmt Syntax" displays if you try to compose a format file. The format file may contain one or many formats.

When setting a job with formats, the format file must be present on either the same disk as the job file or on a disk in the default drive (H1 or F1). If the PowerView 10 cannot file the format file in either place, it displays the message "No File" in the status lines.

Creating A Format File

When entering format definitions, the "syntax" of the format must be correct. Correct syntax means that all parts of the define format command have to be

positioned correctly or an error results and the format cannot be used in the job. The characters used to create a format and the keystrokes required to access them are:

Description	Symbol	Keystrokes
Half intensity open brace	{	PREC UC or PREC LC TYPE CMO
Half intensity close brace	}	PREC UC or PREC LC TYPE CMX
Half intensity open parenthesis	(	PREC LC g
Half intensity close parenthesis	)	PREC UC g

To prevent confusion, in all exercises, we have used PREC LC TYPE CMO for the open format brace ({) and PREC UC TYP CMX for the close format brace (}). You may use either the PREC LC or PREC UC key, whichever is most convenient for you.

The Define Format Command

Suppose you want to create a format in a file named "jobfmt." The text file is very long and the fewer characters, the better. The parameters are point size of 10, linespace of 12, font 81, and line length of 30. Your format will look like this:

{.DFMT,start(<PS10<LS12<FT81<LL30>)}

The define format command consists of a half intensity open brace and .DFMT which must be upper case.

The format name may consist of up to eight alphanumeric characters. Spaces or punctuation cannot be used.

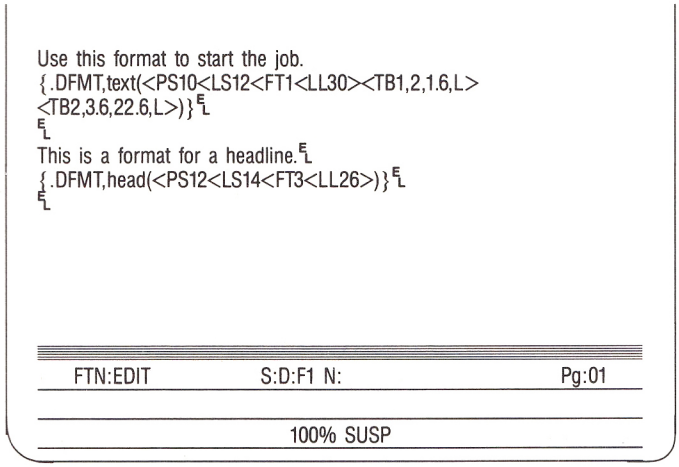
The format definition always begins with a half intensity open parenthesis and may contain typesetting commands, text, format variables, repeat strings, counters, and calls to other formats. The definition must end with a half intensity close parenthesis followed by a half intensity close brace.

Each format can contain as many as 255 characters. You can define as many formats as you like in a format file, but the file cannot exceed 2,000 characters. Only the characters between the open and close braces are considered to be part of the 2,000 character limit. Therefore, you can do some things to make your format files easier to work with.

Put some returns or end lines between the define format comands to separate them. This makes the screen easier to read.

It's also a good idea to include notes within the file to alert you and other operators of any special requirements. The PowerView 10 only pays attention to the characters between the brackets. Extra characters outside of the brackets do not affect format operation.

A format file with comments and end line commands between formats:



Exercise 1—Creating a Format File

You are typesetting a job using the same parameters repeatedly. You decide a good way to save time is to put them in a format file.

You Are Going To:

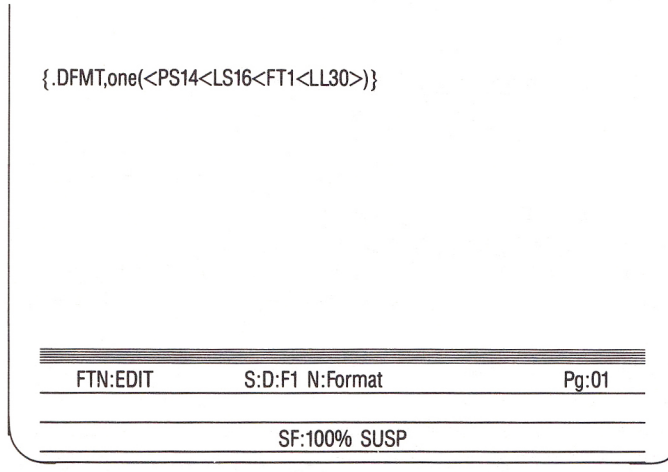
1. Create a format file called "Format".
2. Define a format called "one" using the following parameters:

Point Size	14
Linespace	16
Font	1
Line Length	30

Do The Following:

- Open a new file called *Format*:
  NEW FILE Format EXECUTE
- Make sure your PowerView 10 is in Edit Mode:
  FILE FUNC EDIT EXECUTE
- Define a format called "one" using the following keystrokes:
  PREC LC TYPE CMD .DFMT,one PREC LC g<PS14<LS16<FT1<LL30>
  PREC UC g PREC UC TYPE CMX

Your source window will look like this:



- Before you append and close the file, read the next section.

The Check Format Definitions Function

To ensure that you have entered the Define Format commands correctly, you can have the PowerView 10 check them for you. While the PowerView 10 is in Edit mode and the format definitions are on your screen, press:

FILE FUNC CHKFD EXECUTE

When this command is entered, the cursor will automatically go to the beginning of the file and then pass through each format.

If a syntax error is found, the PowerView 10 positions the cursor at the end of the format containing the error and displays "Fmt Syntax". Make the necessary corrections in the syntax and replace the page. Make sure your PowerView 10 is in Edit mode or an "Invalid Command" message displays.

In Exercise 1, you created a format file. In Exercise 2, you will check it to make sure the syntax is correct.

Exercise 2—Checking The Format

You Are Going To:

1. With the format file "Format" still on the screen,
2. Check the format "one" to make sure the syntax is correct.

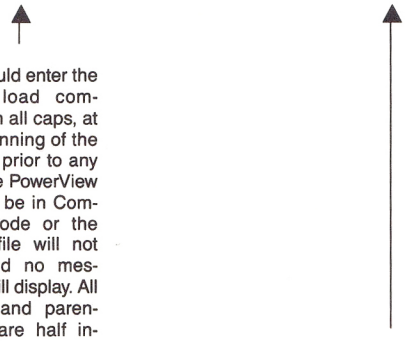
Do The Following:

- ► Verify the accuracy of your formats:
  FILE FUNC CHKFD EXECUTE
- ► If an error is found, correct your syntax.
- ► Append the page and close the file.

Loading a Format File

You have defined and stored a format in a format file. To use the format, you must load the format file into memory for use in a text file. The load format file command is described below.

{.LFD(format filename)}



You should enter the format load command, in all caps, at the beginning of the text file, prior to any text. The PowerView 10 must be in Compose mode or the format file will not load and no messages will display. All braces and parentheses are half intensity.

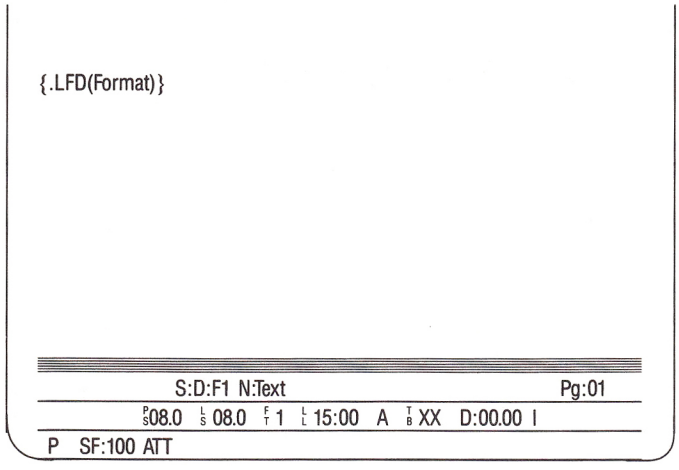
This is the name of the format file already stored on disk. The format file must be stored on the same disk as the text file or in F1 or H1. If the file is stored in a sub-index, the entire pathname must be entered.

Only one format file can be loaded in a text file. If you try to load another in the same text file, the message "Fmt Loaded" will display. If this happens, delete the second .LFD command.

The format file is loaded when the PowerView 10 is in compose mode and the cursor is moved past the load command's closing bracket. If a format file is loaded and you need to use a different one, close the text file, reopen it, and delete the format file before the PowerView 10 can load it. Load the new file and continue as usual.

Format files can be loaded into new or existing text files.

After you have loaded a format file and before you have entered text, your source window will look like this:



"Format" is the name of the format file loaded into the text file.

In Exercises 1 and 2, you created a format file, defined a format and checked the accuracy of the format. The next step is to load that format file into your text file.

Exercise 3—Loading a Format File

You Are Going To:

1. Start a new text file called "Text".
2. Load the format file "Format" into memory for use in the text file.

Do The Following:

- ► Start a text file called "Text":
  [NEW FILE] Text [EXECUTE]
- ► If not in Compose mode:
  [COMPOSE] [EXECUTE]
- ► Load the format file "Format":
  [PREC LC] [TYP CMO] .LFD [PREC LC] g Format [PREC UC] g [PREC UC] [TYP CMX]
- ► Append the page and close the file.

Notice that you have loaded only the format file (Format) into your text file. A format will be called in the next exercise.

Possible Error Messages

FMT SYNTAX
If you make an error while entering the load command in the text file, you will receive the message "Fmt Syntax" and the format file will not load. Check the load command syntax. Make sure .LFD is entered in all caps and both the opening and closing half-intensity brackets and parenthesis are correct.

NO FILE
If the PowerView 10 cannot find the format file, it displays the "No File" message. Make sure the format file is on the same disk as the text file or is on the default disk in F1 or H1. If you are using a hard disk, make sure the sub-index or sub-directory names are correct.

Calling A Format

Once the format file is loaded, you may "call" any format in that file to use in the text file.

Exercise 4—Calling A Format

You are now ready to call the format "one" into your text file and enter copy.

You Are Going To:

1. Call the text file "Text" to the screen. The format file "Format" is already loaded in your text file.

2. Call the format "one" into the text file following the load format file command.

Do The Following:

► Call the file "Text" to the screen:

[CALL PAGE] Text [EXECUTE]

► Move the cursor to the end of your file:

[LIMIT] [SCRL UP]

► Call the format "one" into the text file after the loaded format file:

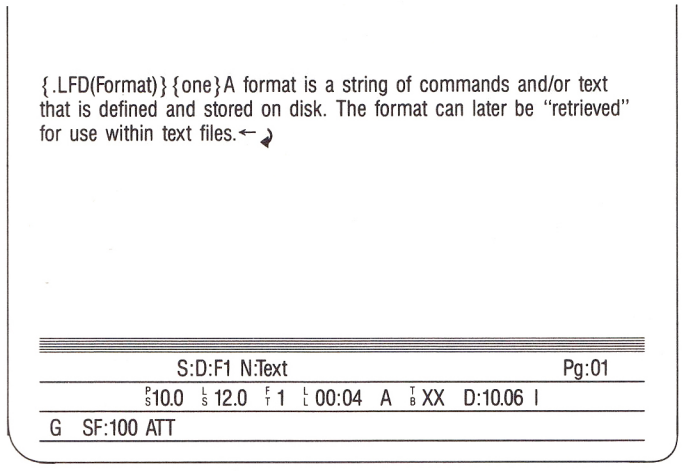
[PREC LC] [TYPE CMO] one [PREC UC] [TYPE CMX]

► Enter the following text in your text file:

A format is a string of commands and/or text that is defined and stored on disk. The format can later be "retrieved" for use within text files.

[LEFT] [RETURN]

The source window of your PowerView 10 will look like this:



The source window shows only the text you type and the format name. The view window shows the job as it will look when it is typeset. Characters in the format file will appear in the view window.

The view window of your PowerView 10 might look like this:

A format is a string of commands and/or text that is defined and stored on disk. The format can later be "retreived" for use within text files.

► Replace the page and close the file:

REPL PAGE EXECUTE CLOSE FILE EXECUTE

Possible Error Messages:

FMT SYNTAX
If a syntax error is found in the format definition, the cursor is positioned on the closing brace (}) and the message 'Fmt Syntax" is displayed. Close the text file and call the format file to the screen. Use the check format function to double-check the format syntax. Make any necessary corrections, replace the format file and close it. Call the text file back to the screen.

NO FORMAT
When a format that does not exist in the loaded format file has been called, the PowerView 10 displays the message "No Format."

FT?? ERROR, PS?? ERROR
If the format contains parameter errors (e.g., no font, incorrect point size, etc.), the PowerView 10 displays the appropriate message (FT?? Error, PS?? Error).

Exploding A Format

You may explode a format name in the text file to display its contents. This is useful for checking and correcting errors within the text file. It is also useful is you want most, but not all, of the characters in a format.

To explode a format in an open text file, make sure the PowerView 10 is in Compose mode. Position the cursor on the closing brace of the format call (}) and press:

FILE FUNC EXFMT EXECUTE

The contents of the specified format will replace the format call.

Exercise 5—Exploding A Format.

You Are Going To:

1. Call the file "Text" to the screen.
2. Explode the format "one".

Do The Following:

► Call "Text" to the screen:

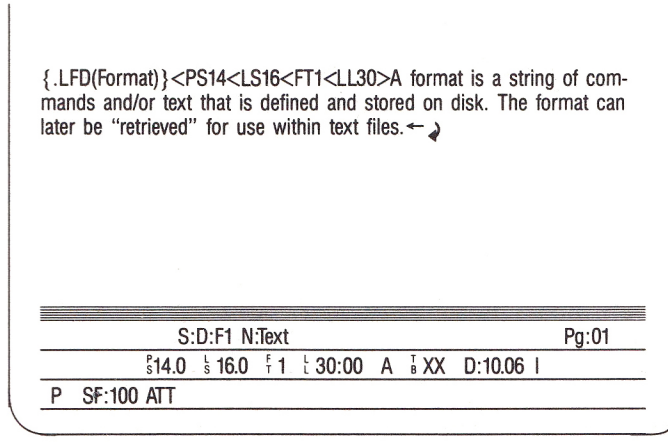
CALL PAGE Text EXECUTE

► Move the cursor to the closing brace of the format call "one".

► Make sure your PowerView 10 is in Compose mode and enter the explode format command:

FILE FUNC EXFMT EXECUTE

The source window of your PowerView 10 will look like this:



The format name "one" was replaced with the contents of format "one" when exploded in the text.

► Close the file. When you call the text file again, the format name "one" will appear in the file, not the exploded format because you did not replace the page.

Summary: Creating and Using a Format

You have learned how to:

- Create a format file.
- Check the syntax of your formats.
- Load the format file into a text file.
- Call a format in a text file.
- Explode a format in the text stream.

Complete the following exercise (Exercise 6) to make sure you understand what has been discussed so far. Detailed instructions are given to help you.

Exercise 6—Creating and Using A Format

In this exercise, you are going to create a format file and then use the format in a text file. Your file will include a bold head and numbered, indented paragraphs.

The view window of your PowerView 10 will look like this:

What Is A Format?

A format is a string of commands and/or text which is defined and stored on disk and can later be "retrieved" for use within text files.

Every format may contain up to 255 characters and has two essential parts, consisting of:

1. Name—from 1 to 8 alphanumeric characters.
2. Definition—including typesetting parameters, text characters, calls to other formats or variable information determined at a later time.

You Are Going To:

1. Create a format file.
2. Open a text file.
3. Load the format file into the text file.
4. Call the format.
5. Enter text in the text file.
6. Look at the view window display.

Do The Following:

- ► Create a new format file called "form.file".
- ► Your PowerView 10 should be in Edit mode:
  FILE FUNC EDIT EXECUTE

► Enter the format definition for the head:

PREC LC TYPE CMO .DFMT,fmt1 PREC LC g<PS12<LS14<FT3<LL28.6>What Is A Format?

LEFT RETURN

PREC UC g PREC UC TYPE CMX RETURN

► Enter the format definition for the text:

PREC LC TYPE CMO .DFMT,fmt2 PREC LC g<PS10<LS12<TB1,,1,L>

<TB2,1,27.6,J> PREC UC g PREC UC TYPE CMX RETURN

► You may want to check the syntax:

FILE FUNC CHKFD EXECUTE

► If there are syntax errors, correct the format and recheck the syntax.

► Append the page and close the file.

► Create a text file called "example".

► Make sure your PowerView 10 is in Compose mode.

► Load the format file:

PREC LC TYPE CMO .LFD PREC LC g form.file PREC UC g PREC UC TYPE CMX

► Enter the format call for the head:

PREC LC TYPE CMO fmt1 PREC UC TYPE CMX

► Enter the format call for the text:

PREC LC TYPE CMO fmt2 PREC UC TYPE CMX

► Enter the following text in your file:

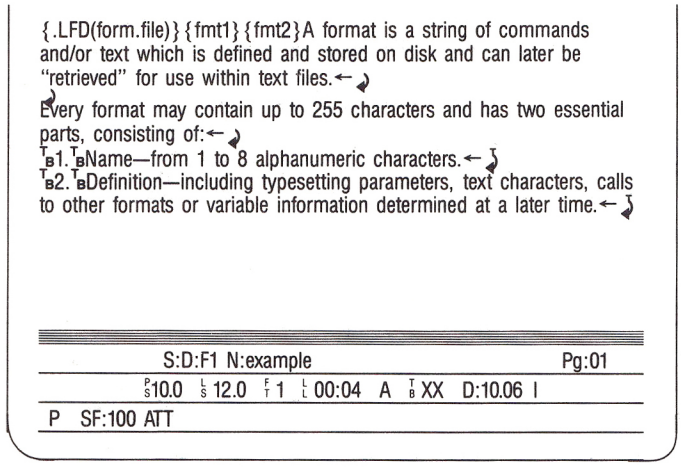
A format is a string of commands and/or text which is defined and stored on disk and can later be "retrieved" for use within text files. LEFT RETURN RETURN

Every format may contain up to 255 characters and has two essential parts, consisting of: LEFT RETURN

TAB 1. TAB Name—from 1 to 8 alphanumeric characters. LEFT TAB RET

TAB 2. TAB Definition—including typesetting parameters, text characters, calls to other formats or variable information determined at a later time. LEFT TAB RET

The source window of your PowerView 10 will look like this:



- Any errors in your format will display on the message line.
- As you type, the text will appear in the view window as well as the source window. Compare your view window with the sample on page IV-17.
- Append the page and close the file.

Format Variables

The formats discussed so far cannot be altered when being used in a job file. Each format has a fixed definition. If you want to change anything in the format while you are setting a job, you must explode the format or alter the format file.

With format variables, you can alter the contents of a format while you are working in the job file, without exploding the format. A variable can be thought of as an "open area" in a format definition. These open areas can be filled when the format is called in the job.

A format definition with variables looks like this:

{.DFMT,head,,(<PSⒹ1<LSⒹ2<FT3<LL20>)}

The commas after the format name "head" indicate that there are two variables. The "Ⓓ1" and "Ⓓ2" in the format definition indicate where the variable information will be placed when the format is called. Each data symbol must be followed by a number from 1-9. The number refers to the comma's position. Information following the first comma will be placed where Ⓓ1 is; information following the second comma will be placed where Ⓓ2 is. Each comma following the format name must have at least one data symbol with the appropriate position number in the format definition. For example, if you have three commas, you must have Ⓓ1, Ⓓ2 and Ⓓ3 in the format definition.

As many as nine variables can be used in a format (nine commas with nine numbered data symbols). All variables can contain a total of up to 255 characters when the format is called. A variable value may be a letter or number, a command, or even a format call.

To use a format with variables in a text file, you *must* put the variable information in the format call. For example:

{head,18,24}

The number 18 will be placed where Ⓓ1 is and the 24 will be placed where theⒹ2 is. In this example, the point size will be 18 and the linespace will be 24. If you wanted the point size to be 14 and the linespace to be 15, you would enter the format call like this in the text file:

{head,14,15}

Exercise 7—Variables in Formats

Suppose you want to use the same format that was just defined, but you want to change the size and style of the headline continually. You can use three variables; one for the size, the second for the style and the third for the headline itself.

You Are Going To:

1. Open a new format file.
2. Define a format with three variables.
3. Create a new text file.
4. Call the format into the file, filling in the variable positions.

Do The Following:

► Open a new foramt file called "variable".

► Define a format called "head" in which there are three variables:
[PREC LC] [TYPE CMO] .DFMT,head,,, [PREC LC] g<PS [DATA] 1<FT [DATA] 2> [DATA] 3
[LEFT] [RETURN] [PREC UC] g [PREC UC] [TYPE CMX] [END LINE]

► Check the format syntax and correct any errors, if necessary.

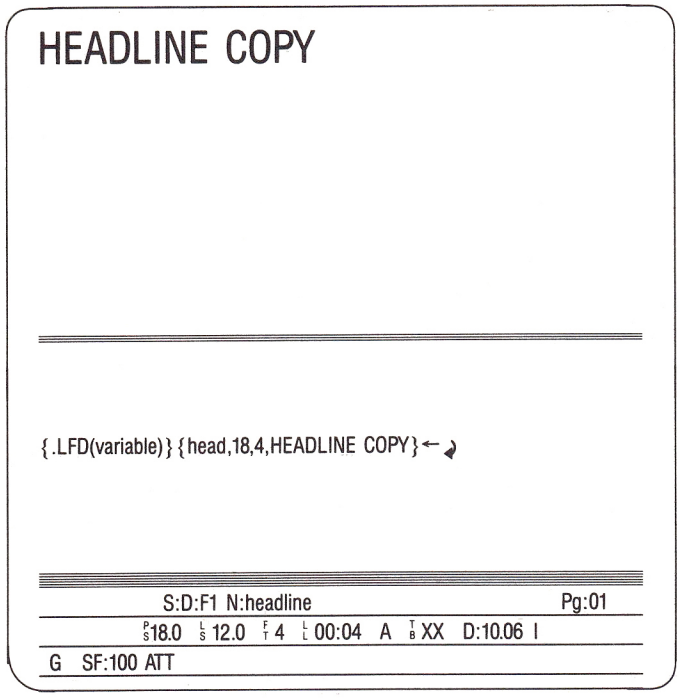
► Append the page and close the file.

► Open a text file called "headline".

► Load the format file "variable":
[PREC LC] [TYPE CMO] .LFD [PREC LC] g variable [PREC UC] g [PREC UC] [TYPE CMX]

► Call the format "head" into the text file and fill in the variable positions:
[PREC LC] [TYPE CMO] head,18,4,HEADLINE COPY [PREC UC] [TYPE CMX]

PowerView 10 screen will look like this:



► Append the page and close the file.

Remember that what is inserted following the first comma in the default area becomes the contents of Ⓓ1. Data following the second comma becomes the contents of Ⓓ2, etc., up to nine variables.

Escape Characters

In some formats, certain characters function as part of the format syntax *and* as punctuation in the format. To use the colon (:) as a punctuation mark in any part of a format definition, or the comma as punctuation in the variable defaults, they must be preceded with another colon. The first colon is an "escape" character. An escape character indicates that the colon or comma following it will be typeset. If in doubt whether to escape a particular character, precede it with a colon.

Complete Exercise 8 to make sure you understand the procedure used to escape a character. In addition, this exercise will test your understanding of entering text as a variable in a format string.

Exercise 8—Using Text as a Variable and Using Escape Characters

You are typesetting form letters; the change in each letter is the name of the addressee. A format can be set up using a variable that will allow you to do this. In addition, you must enter an escape colon preceding the colon that will be typeset.

The text for the form letter is also included in the format; therefore, you need to change only the addressee's name every time you call this format into your text file. You do not have to re-enter the text for the letter. The format calls with the different names entered in the variable position will be the only codes and text that appear in the text file.

You Are Going To:

1. Open a new format file called "letterfmt".
2. Define a format called "letter" containing a variable that allows you to change the salutation in a form letter. The format also contains an "escape" colon and the text for the form letter.
3. Open a new text file called "heat" in which the format is used.

Do The Following:

- ► Open a new format file named "letterfmt".
- ► Put your PowerView 10 in the Edit mode:
  [FILE FUNC] EDIT [EXECUTE]
- ► Define a format called "letter".
  [PREC LC] [TYPE CMO] .DFMT,letter, [PREC LC] g<PS12<LS14<FT1<LL20>
  Dear [DATA] 1:: [LEFT] [RETURN]
  CUT YOUR HEATING BILLS THIS WINTER! [LEFT] [RETURN]
  [RETURN]
  This foolproof method of heating your home will save you a great deal of money. This new technique was developed in Europe. [LEFT] [RETURN]
  [RETURN]
  Call me at the number listed below. [LEFT] [RETURN]
  [PREC UC] g [PREC UC] [TYPE CMX]

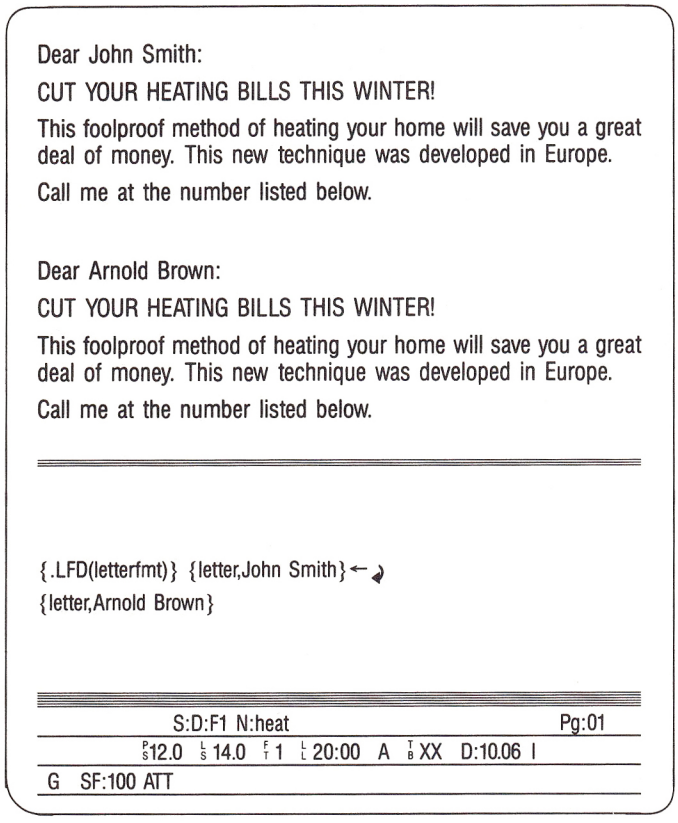
The comma following the format name "letter" references DATA 1 in the format definition. Note also the escape colon.

- ► Check the format syntax and make any necessary corrections.
- ► Append the page and close the file.

- ► Open a new text file named "heat".
- ► Make sure your PowerView 10 is in Compose mode.
- ► Load the format file "letterfmt":
  PREC LC TYPE CMO .LFD PREC LC g letterfmt PREC UC g PREC UC TYPE CMX
- ► Call the format "letter" and enter the variable "John Smith":
  PREC LC TYPE CMO letter,John Smith PREC UC TYPE CMX RETURN
- ► Call the format "letter" again and enter the variable "Arnold Brown":
  PREC LC TYPE CMO letter,Arnold Brown PREC UC TYPE CMX RETURN

The "John Smith" and "Arnold Brown" you enter after the format name are variables. They can be changed every time you want to send this form letter.

The PowerView 10 screen will look like this:



► Append the page and close the file.

Format Variables with Defaults

When defining variables in a format definition, you can also assign a "default" value for each variable. The defaults will be used unless you enter a different variable when you call the format.

To assign defaults to a format variable, enter the values after the format name as you define the format. If you enter a comma without a value when you call the format in a text file, the PowerView 10 will use the default value.

Suppose you define a format like this:

{.DFMT,head,14,3,HEADLINE COPY(<PS Ⓓ 1<FT Ⓓ 2>Ⓓ 3)}

The default values for this format are point size 14, font 3 and the words HEADLINE COPY. If you enter the format in the text file like this:

{head,,,}

the words HEADLINE COPY will appear in 14 point, font 3, type on your PowerView 10 view window screen.

You can override the default by entering a value for the variables when you call the format. If you enter the format in the text file like this:

{head,24,,}

the words HEADLINE COPY will appear in 24 point, font 3, on your PowerView 10 view window screen. When you typed the number 24, you overruled the default of 14. If you want a point size of 14, font 3 and the words MARTIN TO ENTER PRESIDENTIAL PRIMARY, you would type:

{head,,MARTIN TO ENTER PRESIDENTIAL PRIMARY}

Format Nesting

So far, each format has been used to call a group of characters. Formats can call other formats as well. When a format is used to call another fomat, it is referred to as nesting.

Nesting formats can expand the 255 format character limit by linking two or more formats together as if they were one. Nested formats can also save keystrokes. If two or more formats are often used together, you can nest them and call all of them with one format.

You can nest as many as eight formats within one format. If you try to call a format which has more than eight nest levels, the PowerView 10 will give you a "Nest Error" message.

Exercise 9—Nesting Formats

The easiest way to learn about nested formats is to see a simple example.

In the following example, you will see how one format calls another.

You are going to:

1. Open a new format file.
2. Define a format with four nested formats.
3. Open a new text file.
4. Load the format file and call the format.
5. Observe what happens on the view window.

Do the Following:

► Open a file called fmtnest.

► Define a format called one:

PREC UC TYPE CMO .DFMT,one PREC LC TYPE CMO g<PS12<LS14<FT1<LL24>This is format ONE. This format calls format two. RETURN {two}
The end. QUAD LEFT RETURN PREC LC g PREC LC TYPE CMX

► Define a format called two:

PREC UC TYPE CMO .DMFT,two PREC LC TYPE CMO g<PS8>This is two. Two calls the format called three. RETURN RETURN {three} PREC LC g PREC L TYPE CMX

► Define a format called three:

PREC UC TYPE CMO .DMFT,three PREC LC TYPE CMO g<PS24>This is format three. It calls format four. RETURN {four} PREC LC g PREC L TYPE CMX

► Define a format called four:

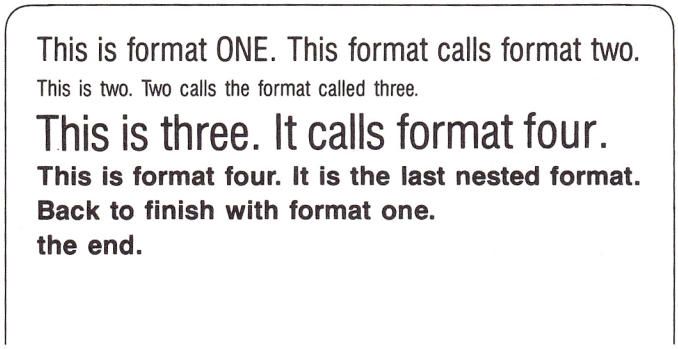
PREC UC TYPE CMO .DMFT,four PREC LC TYPE CMO g<PS10<FT3>This is format four. It is the last nested format. Back to finish with format one. RETURN PREC LC g PREC L TYPE CMX

► Append and close the file.

► Open a text file called nest.

► Load the format file "fmtnest" and call the format "one."

The view window of your PowerView 10 will look like this:



Exercise 10—Nesting and Format Variables

You are going to call the form letter format file you made earlier to the screen and nest another format inside it. You will call the form letter text file you made earlier and examine the view window to see how the nested format affects the job.

You Are Going To:

1. Call the file "letterfmt" to the screen. This file was created in Exercise 8.
2. Nest a format at the end of the format in the file.
3. Define a new format.
4. Replace the page and close the file.
5. Call the text file "heat" to the screen and examine the text in the view window.

Do The Following:

- ► Call the file "letterfmt" to the screen.
- ► Position the cursor on the closing half intensity parenthesis of the first format.
- ► Nest the format name "phone" at the end of the format in the file: [PREC LC] [TYPE CMO] phone [PREC UC] [TYPE CMX].

► Define a new format called "phone":

PREC LC TYPE CMD .DFMT,phone PREC LC g 777-9999 RIGHT RETURN

EUROPEAN HEATING RIGHT PREC UC g PREC UC TYPE CMX END LINE

► Replace the page and close the file.

► Open the text file "heat." Move the cursor to the end of the job and compose.

The PowerView 10 screen will look like this:

Dear John Smith:

CUT YOUR HEATING BILLS THIS WINTER!

This fool-proof method of heating your home will save you a great deal of money. This technique was developed in Europe.

Call me at the number listed below.

777-9999
EUROPEAN HEATING

Dear Arnold Brown:

{.LFD,letterfmt}{letter,John Smith}{letter,Arnold Brown}

S:D:F1 N:heat Pg:01

P S 12.0 L S 14.0 F T 1 L L 20:00 A T B XX D:10.06 I

G SF:100 ATT

Using a Variable More Than Once

A variable may be used several times in the same format definition. To do this, use the same numbered data symbol wherever you want the same information.

Suppose you want to set this job:

Compugraphic™Corporation

The coding looks like this:

<PS18>Compugraphic<PS8<RLD6>TM<ALD6<PS18> Corporation

The point size 18 is repeated two times. The add/reverse lead 6 is repeated two times. A format that used these two vaules as variables would look like this:

{.DFMT,cg,18,6(<PSⒹ1>Compugraphic<PS8<RLDⒹ2>TM<ALDⒹ2<PSⒹ1>Corporation)}

Each variable is used twice in the format definition. You can use any variable as often as you like in a format definition and in any order (Ⓓ2 can appear before Ⓓ1, for example).

Passing Variables To Nested Formats

This process combines nesting and variables. The variable you enter in your format call will act as a variable in *another* nested format call inside the format definition.

Exercise 11—Passing Variables To Nested Formats

You have to set this weekly announcement:

NOW PLAYING
FOOTSIE
AT THESE SELECTED THEATRES
ANDOVER
BOSTON
CHELMSFORD
HAVERHILL
LAWRENCE
MALDEN
NORTH READING
PEABODY
QUINCY
FOOTSIE
Call 888-7777 FOR TIMES

The title "FOOTSIE" changes each week. The title will be a format variable. As you set up the format definition, the theatre listing in the format takes you over the 255 character limit. You decide to make two smaller formats and nest the second format call into the first one.

You realize one variable (the movie title) is in the second format. You could call two formats into your job, but you want to use only one format.

You Are Going To:

1. Open a new file.
2. Define a format that includes variables and nests a second format.
3. Define a second format.
4. Append the page and close the file.
5. Open a new text file.
6. Call the format into the text file.
7. Fill in the variable positions.
8. Examine the view window.
9. Append the page and close the file.

Do The Following:

- Open a new file called "listings".
- Define a format called "name" that includes variables and nests a second format called "time":

  [PREC LC] [TYPE CMD] .DFMT,name, [PREC LC] g<AC<PS18<LS18<FT3><KO>
  NOW PLAYING [RETURN]
  <PS16<LS16> [DATA] 1 [RETURN]
  <FT1>AT THESE SELECTED THEATRES<LS10> [RETURN]
  <PS10>ANDOVER [RETURN]
  BOSTON [RETURN]
  CHELMSFORD [RETURN]
  HAVERHILL [RETURN]
  LAWRENCE [RETURN]
  MALDEN [RETURN]
  MEDFORD [RETURN]
  NORTH READING [RETURN]
  PEABODY [RETURN]
  QUINCY [RETURN]
  [PREC LC] [TYPE CMD] time, [DATA] 1 [PREC UC] [TYPE CMX] [PREC UC] g [PREC UC] [TYPE CMX] [END LINE]
  [END LINE]

- Define a second format called "time":

  [PREC LC] [TYPE CMD] .DFMT,time, [PREC LC] g<ALD24<PS24<FT3> [DATA] 1<LS10> [RETURN]
  <PS8<FT1>Call 888-77 77 FOR TIMES [RETURN]
  [PREC UC] g [PREC UC] [TYPE CMX] [END LINE]
  [END LINE]

- ► Append the page and close the file.
- ► Open a text file called "theatres".
- ► Load the format file "listings":

  PREC LC TYPE CMO .LFD TYPE CMO g listings PREC UC g PREC UC TYPE CMX

- ► Call the format "name" and fill in the variable position:

  PREC LC TYPE CMO name,FOOTSIE PREC UC TYPE CMX

- ► Look at the text in the view window. It should resemble the announcement on page IV-33.
- ► Append the page and close the file..

Counters

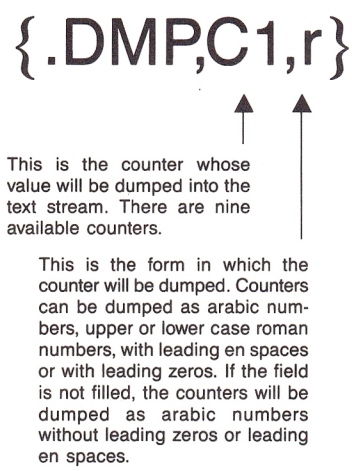
Counters are values that the PowerView 10 can increment each time it uses them. Counters can be set equal to any value from 0 to 32,767. They can be added to, subtracted from, multiplied by or divided by a fixed amount each time they are used. On of the most common uses of counters is automatic page numbering.

Counters can be used in both the format and text files. They cannot be part of a format name or part of a typesetting command in a format.

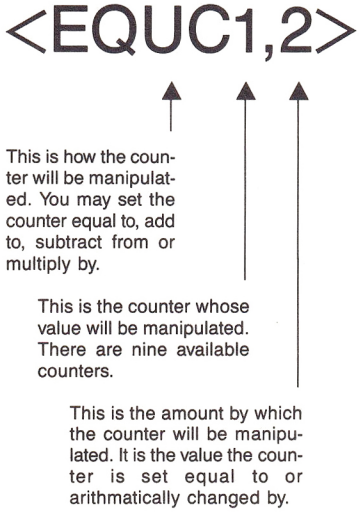
Nine counters (numbered 1 to 9) are available for use within each file. Each counter works independent of the others. Changing the value of one will not necessarily affect the others.

A counter is not typeset or displayed until it is "dumped." You must enter a dump command each time you want the counter to appear in the text. The current value of the counter is displayed in the Paramter Display.

The dump command looks like this:



Counter values must be set before a counter can be dumped. Counters can be set equal to a value (EQU), have a value added to them (ADD), have a value subtracted from them (SUB), and be multiplied or divided by a value (MUL), (DIV). The counter manipulation command looks like this:



Exercise 12 — How Counters Work

To see how counters work, do the following:

► Set Counter 1 equal to 9:

[TYPE CMO] EQUC1,9 [TYPE CMX]

► Dump Counter 1:

[PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] [RETURN]

☛ **NOTE:** *The counter designation (C#) in a dump command must always be an upper case letter.*

► Set Counter 2 equal to 1:

[TYPE CMO] EQUC2 [TYPE CMX]

► Dump Counter 2:

[PREC LC] [TYPE CMO] .DMP,C2 [PREC UC] [TYPE CMX] [RETURN]

☛ **NOTE:** *When no value is entered following the counter number in the counter command, it will default to 1.*

► Add the value of Counter 2 to the value of Counter 1:

[TYPE CMO] ADDC1,C2 [TYPE CMX]

► Dump Counter 1:

[PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] [RETURN]

► Subtract the value of Counter 2 from the contents of Counter 1:

[TYPE CMO] SUBC1,C2 [TYPE CMX]

►Dump Counter 1:

[PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] [RETURN]

► Multiply the value of Counter 1 by 5:

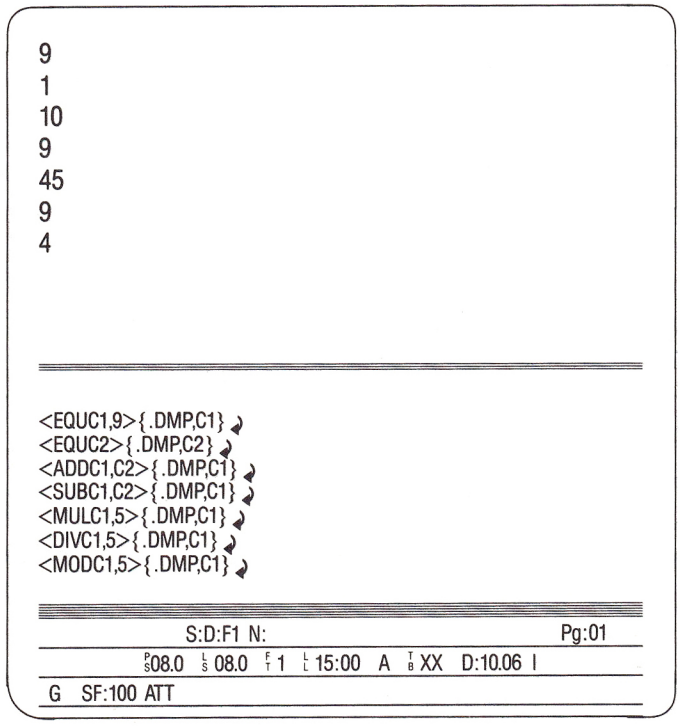
[TYPE CMO] MULC1,5 [TYPE CMX]

► Dump Counter 1:

[PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] [RETURN]

- Divide the value of Counter 1 by 5:
  [TYPE CMO] DIVC1,5 [TYPE CMX]
- Dump Counter 1:
  [PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] [RETURN]
- Divide the value of Counter 1 by 5 and store the remainder in Counter 1:
  [TYPE CMO] MODC1,5 [TYPE CMX]
- Dump Counter 1:
  [PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] [RETURN]

The PowerView 10 screen will look like this:



Exercise 13—Using and Dumping Counters

You need to typeset a list of numbered heads for an outline.

You Are Going To:

1. Start a new file called "counterfmt".
2. Add a format defining and dumping a counter.
3. Replace the page and close the file.
4. Start a new file called "heads".
5. Load the format file.
6. Enter job parameters and call the format "num" before each head. The numbers will automatically increment.
7. Append the page and close the file.

Do The Following:

► Start a new file called "counterfmt".

► Define a format called "num" which defines and dumps a counter:
[PREC LC] [TYPE CMO].DFMT,num [PREC LC]g<ADDC1,1> [PREC LC] [TYPE CMO].DMP,C1 [PREC UC] [TYPE CMX]
[PERIOD] [THIN SPACE] [PREC UC]g [PREC UC] [TYPE CMX] [RETURN]

► Append the page and close the file.

► Start a new text file called "heads".

► Load the format file:
[PREC LC] [TYPE CMO].LFD [PREC LC]g counterfmt [PREC UC]g [PREC UC] [TYPE CMX]

► Enter the following job parameters:
Point size 10
Linespace 12
Font No. 1
Line length 20
Automatic Quad Left
Kerning On

► Enter the format call "num" for the first section head:
[PREC LC] [TYPE CMO]num [PREC UC] [TYPE CMX]The Production Cycle [RETURN]

► Enter the remaining section heads, each preceded by a format call

to format "num".

PREC LC TYPE CMO num PREC UC TYPE CMX Calling the File RETURN
PREC LC TYPE CMO num PREC UC TYPE CMX Editing the File RETURN
PREC LC TYPE CMO num PREC UC TYPE CMX Screen Modes RETURN
PREC LC TYPE CMO num PREC UC TYPE CMX Replacing the File RETURN
PREC LC TYPE CMO num PREC UC TYPE CMX Typesetting the File RETURN

The PowerView 10 screen will look like this:

1. The Production Cycle
2. Calling the File
3. Editing the File
4. Screen Modes
5. Replacing the File
6. Typesetting the File

{.LFD(counterfmt)}<PS10<LS12<FT1<LL20<AL>
{num}The Production Cycle
{num}Calling the File
{num}Editing the File
{num}Screen Modes
{num}Replacing the File
{num}Typesetting the File

S:D:F1 N:heads Pg:01
PS10.0 LS 12.0 FT 1 LL 20:00 A TB XX D:10.06 I
G SF:100 ATT

► Append the page and close the file.

Qualifying Counter Output

In the last example, the counters were dumped as arabic numerals. A qualifier may be added to the .DMP command to change the way that counter is dumped. The qualifier is added to the dump command after the number of the counter to be dumped:

{.DMP,C1,R}

The qualifiers and their functions are:

Function	Qualifier
Suppress leading zeroes	S (default)
Output Roman numerals	Upper (R) and lower case (r) (May not exceed 3,999)
Output leading zeroes	(L#) Total # of digits including counter value (maximum: 9)
Output leading EN spaces	(N#) Total # of digits including counter value (maximum:9)

All counters will output as arabic numbers unless you use the qualifiers R or r

Example:

The following counter is defined to output as 5. Dumping it without a qualifier will output an arabic number:

<EQUC1,5>	{.DMP,C1}	5

Adding the qualifier "R" displays the 5 as an upper case roman numeral:

<EQUC1,5>	{.DMP,C1,R}	V

Using the qualifier "r" displays the 5 as a lower case roman numeral:

<EQUC1,5>	{.DMP,C1,r}	v

Using the qualifier "L6" displays the 5 with 5 zeroes in front of it: (The 5 takes the place of one of the leading zeroes.)

<EQUC1,5>	{.DMP,C1,L6}	000005

Using the qualifier "N4" displays the 5 with 3 en spaces in front of it: (The 5 takes the place of one of the en spaces.)

<EQUC1,5>	{.DMP,C1,N4}	5

Using The Dump Command to Output a Value

The dump command may also be entered in a text file to output a value not defined in a counter. For example, you want to output the year 1986 in roman numerals. Your dump command will look like this:

{.DMP,1986,R}

Your output will look like this:

MCMLXXXVI

Repeat Command

The repeat command allows you to produce a string of characters from 1 to 999 times. The repeat string can contain text, typesetting commands and/or format calls and can appear in format definitions or in text. The repeat command may contain up to 255 characters.

{.RPT,repeat count(repeat string)}

The Repeat command consists of an {(open brace) and .RPT, and must begin every repeat command.

The repeat count indicates how may times the repeat string will reproduce from 1 to 999 times.

The repeat string must begin with an ((open parenthesis) and may contain text, typesetting commands and/or format calls. It can appear in format definitions or in text and must end with a close parenthesis ()) and a close brace (}).

Exercise 13 will show you how to set up a repeat (.RPT) command using text in the repeat string.

Exercise 14—Repeating Text

You Are Going To:

1. Open a new file.
2. Enter a repeat command that will output a newspaper head 30 times.
3. Examine the text in the view window.
4. Append the page and close the file.

Do The Following:

- ► Open a new file called "repeats".
- ► Enter these parameters:

  <PS10<LS12<KO>
- ► Enter a repeat command using the following keystrokes:

  [PREC LC] [TYPE CMO] .RPT,30 [PREC LC] g THE ATHOL DAILY NEWS [LEFT] [RETURN]
  [PREC UC] g [PREC UC] [TYPE CMX]

The 30 in the repeat command means that the newspaper headline will repeat 30 times.

- ► Examine the view screen and see the headline repeated 30 times.
- ► Append the page and close the file.

Format Review Exercise

Complete the following review exercise to reinforce what you have learned about formats. Keystrokes are given following the exercise. Try to do the exercise without referring to the keystrokes.

Typeset the following coupon and produce it twice on the page.

Please send 7 issues of FLYING for $6.99 and bill me.
(Full subscription price $9.98)

Print Or Type—Use Only One Card Per Person

Name _____

Address _____

City _____ State _____ Zip _____

(The following is for office use only.)

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____

You will be using these functions already discussed in this section:

1. Define a format.
2. Create a format file.
3. Nest a format.
4. Load the format file.
5. Call a format into the text file.
6. Use counters.
7. Use the repeat function.

Appearing in your text file will be the load format command and the format calls (to the same format) only. Don't forget to enter a command to reset the counter to "0" before the second format call. Otherwise, the numbers in the second coupon will increment, starting with 7, instead of starting at 1.

You Are Going To:

1. Start a new format file.
2. Define a format for the coupon.
3. Start a new text file.
4. Load the format file and call the format.
5. Output the text file.

Do The Following:

- ► Start a format file called "cardfmt".
- ► Define a format called "begin" for the first part of the coupon (through "Zip").
- ► Define and nest a second format called "coupon" into the "begin" format for the remainder of the coupon.
- ► Check the accuracy of the format file.
- ► Append the page and close the file.
- ► Start a text file called "FLYING".
- ► Load the format file.
- ► Call the format "begin" into your text file.
- ► Add lead and reset the counter to 0.
- ► Call the format "begin" into your text file a second time.
- ► Append the page and close the file.
- ► Output the text file.

If you need assistance, refer to the keystrokes given on the next page and in the representations of your format and text files. Remember—this is not the only way to do this job. You may find a method that is easier for you.

- Open a file called "cardfmt":
- Define a format called "begin" for the beginning of the coupon (through "Zip"):
  [PREC LC] [TYPE CMO] .DFMT,begin [PREC LC] g<AL><PS10<LS12>
  <FT1<LL28><KO>Please send 7 issues of FLYING for $6.99 and bill me. [CENTER] [RETURN]
  (Full subscription price $9.98) [CENTER] [RETURN]
  [RETURN]
  Print Or Type—Use Only One Card Per Person [CENTER] [RETURN]
  [RETURN]
  Name [INS SP] [INS BLR] [RETURN]
  [RETURN]
  Address [INS SP] [INS BLR] [RETURN]
  [RETURN]
  City [INS SP] [INS BLR] [INS SP] State [INS SP] [INS BLR] [INS SP] Zip [INS SP] [INS BLR] [RETURN]
  [PREC LC] [TYPE CMO] coupon [PREC UC] [TYPE CMX] [PREC UC] g [PREC UC] [TYPE CMX] [RETURN]
- Define a format called "coupon" for the rest of the coupon. This format is nested into the "begin" format which you have already defined.
  [PREC LC] [TYPE CMO] .DFMT,coupon [PREC LC] g<PS10>(The following is for office use only.) [CENTER] [RETURN]
  [RETURN]
  <LS18> [PREC LC] [TYPE CMO] .RPT,6 [PREC LC] g [INS SP] [INS BLR] [END LINE]
  <ADDC1,1><RLD2> [PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX].
  <ALD2> [LEFT] [RETURN]
  [PREC UC] g [PREC UC] [TYPE CMX] [PREC UC] g [PREC UC] [TYPE CMX] [RETURN]
  [RETURN]

(Continued)

- ► Check the accuracy of your formats:
  [FILE FUNC] CHKFD [EXECUTE]
- ► Append the page and close the file.
- ► Start a file called 'FLYING."
- ► Load the format file "cardfmt" and call the format "begin" into the file twice to repeat the coupon:
  [PREC LC] [TYPE CMO] .LFD [PREC LC] g cardfmt
  [PREC UC] g [PREC UC] [TYPE CMX] [PREC LC] [TYPE CMO] begin [PREC UC] [TYPE CMX] [END LINE] <ALD30>
  <EQUC1,0> [PREC LC] [TYPE CMO] begin [PREC UC] [TYPE CMX] [RETURN]

If you had difficulty completing the previous exercise, review the procedures and try the exercise again. When you feel comfortable using formats, proceed to the Pagination Section.

PAGINATION

Before beginning Pagination, it is important that you complete Sections III and IV, Area Composition and Formats. The Pagination process combines format and area composition commands.

The Pagination Process

By combining formats and area composition tools, it is possible to set a job and have the copy automatically divide into fully laid out pages. To do this, two new commands must be used, the "FOOT" and "HEAD" commands.

These commands are generally found in a format file and are used to call formats automatically when certain conditions are met. For instance, the foot command will call a format when it recognizes that the last compartment on a page is full. The called format will generally enter a page number at the bottom of a page.

The head command will call a format you specify when it recognizes the top of a new page. The called format will generally put in the crop marks for the page, enter a title and set the compartments for copy flow.

When using the foot and head commands, you will see the name of the called formats in half intensity type. The foot format will be followed by a half-intensity <?NP> command. You will see the information in the header and footer displayed in the View Window.

You may edit the job, and add or delete blocks of text. The job will be repaginated automatically.

The PowerView 10 allows one output page per file page. When a page is completely filled and the footer and new page symbols appear, the PowerView 10 will beep and the message "Page Overflow" will appear in the lower right hand corner of the screen. When this happens, you should append or replace the page and continue with on the next file page.

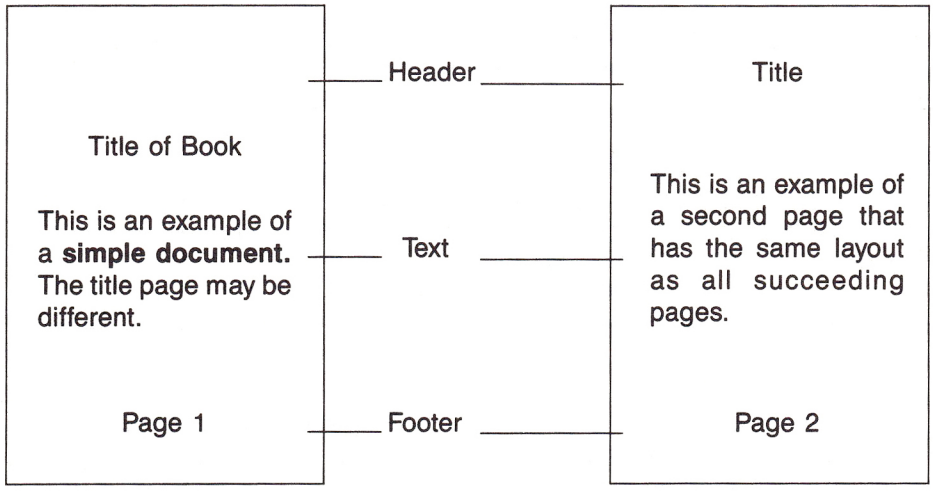
Types of Multiple Page Documents

Three types of multiple page documents can be set using Power Page. Each one can have one or many columns, headings or page numbers. What distinguishes them is not the complexity of the layout, but the *number* of layouts used in each document.

Pages With the Same Layout

In the first kind of multiple page document, each page has the same layout. Examples of this are most paperback novels and reference documents such as dictionaries or parts catalogs.

Even though each page has the same layout, the title page may be different.

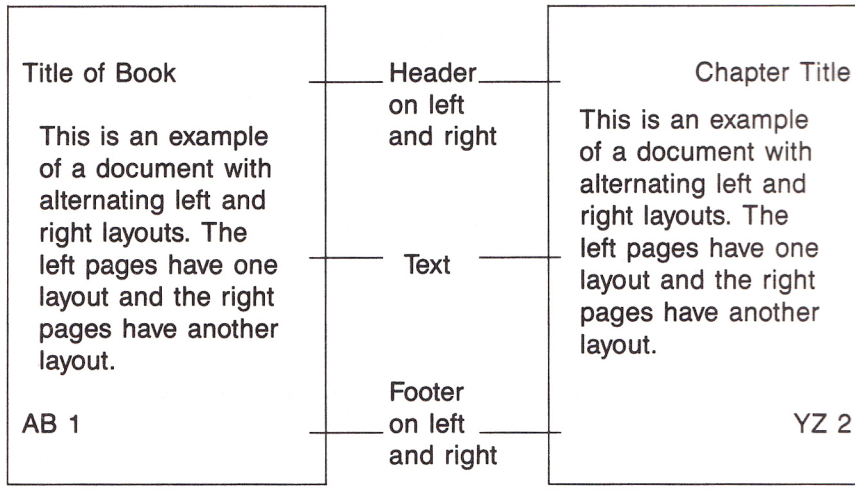


Title or Chapter Page

Pages with Same Layout

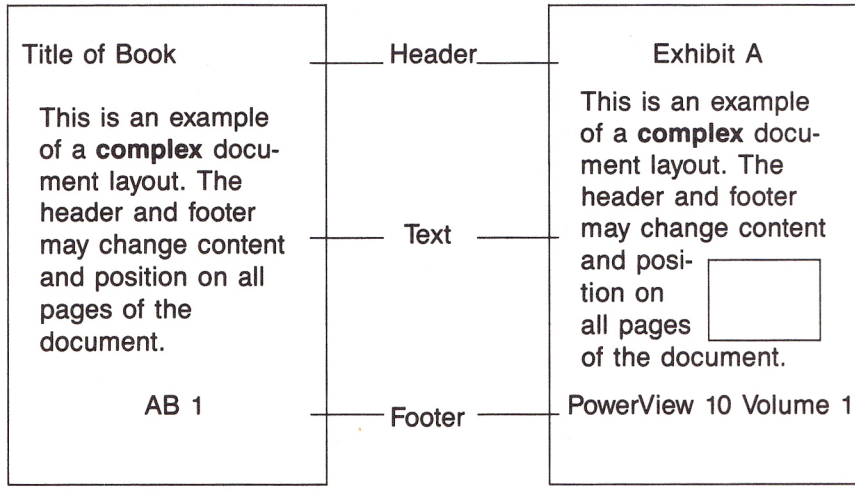
Pages with Alternating Left and Right Layouts

The second kind of multiple page document is one in which all the left pages have one layout and the right pages have another.



Different Layouts on Each Page

The third kind of multiple page document is one in which each page has a different layout, such as a magazine or newspaper. While the other types of layouts can be paginated automatically, this kind must be set *interactively*. You must set the job page by page, entering positioning commands in the text. Depending on the length of the document and the complexity of the layout, even this kind of document can sometimes be paginated automatically.



Simple Multiple Page Documents

In a simple multiple page document, the layout on each page is exactly the same with the possible exception of the title page.

The information for typesetting such a document includes the following:

- A definition of the page size.
- A compartment containing the header information.
- A compartment containing parameters for placement of text on the page.
- A compartment containing the footer information.

The format features of the Power Page option will allow you to store this information in formats.

Setting a Simple Multiple Page Document

Your format files will usually contain three types of formats. One type contains the parameters and layout for the title page. A second type contains the layout for all succeeding pages. The third type contains the "footer" information and includes any text or rules that will appear at the bottom of each page.

Just these formats by themselves will not provide automatic pagination. You must use two new commands, the "FOOT" and "HEAD" commands, in the formats you define.

FOOT COMMAND <FOOT::footer format name::>

The foot command is generally used in the format that contains the layout of the title page and also in the format that contains the layout of each succeeding page. When the bottom of the last compartment is reached, the FOOT command automatically calls the format that contains the footer information. This format, in turn, contains the HEAD command.

You can used the foot and head commands in either the format file or the text file. They are most commonly used in the format file. When used in a text file, the foot command requires only one colon in each location.

HEAD COMMAND <HEAD::header format name::>

The head command is used in the format that contains the footer information. Once the footer information is set, the top of the next page is called and the HEAD command automatically calls the format that contains the page layout. This format, in turn, contains the FOOT command that is activated once the page is full.

The PowerView 10 recognizes when the bottom and top of each page is reached and automatically calls out the appropriate format.

Suppose you have to typeset a paperback book. The title page has a large chapter name centered above the text. All other pages have the title of the book centered at the top of the page and a page number centered at the bottom of the page. You will need the following formats:

1. A format for the title page that contains:
   - A page command and overall job parameters such as Kerning On.
   - Parameters and positioning commands for the chapter name.
   - A compartment for placement of text on the page.
   - A command that calls the footer format when the text compartment is full.

2. A format for the footer that contains:
   - A compartment with a counter command for the page number.
   - A command that calls the header for the next page.

3. A format for the header that contains:
   - Information that will appear on the top of each page (the title of the book, for example).
   - A compartment for placement of text on the page.
   - A command to call the footer format when the text compartment is full.

Once you have set up the formats in a file, you will do the following:

1. Start a text file.
2. Load the format file.
3. Enter a page command.
4. Call the header format.

After you call the title page format, begin entering text.

1. When the text compartment is full, the format automatically calls the footer. You will see the format call for the footer appear in half-intensity type.
2. The half-intensity footer call is followed by a **new page** message. The PowerView 10-generated new page command looks like this: <**?NP**>. This message is a screen prompt to remind you the page has ended.
3. The PowerView 10 will beep and the message "Page Overflow" appears in the lower right hand corner of the screen. You must append or replace the page you have just completed.
4. The footer automatically calls the next header, and the cycle starts over. You will see the header format call in half-intensity type at the top of the new page. Continue typing until you reach the end of this page. Append or replace the page and continue.

The following exercise will give you practice with header and footer commands. The foot command will appear in the title page and header formats. The command tells the PowerView 10 to put in the footer when it needs to. The head command appears in the footer format. When the footer is called, it will automatically call the next header.

You will see the half-intensity footer and header formats and the new page message <?NP> in this exercise. You must append each new output page as a separate file page when the "Page Overflow" message appears.

Exercise 1—Simple Document

Exercise 1 will show you how to set up the formats for an automatically paged simple document.

The following specifications apply to Exercise 1. You have a paperback book to typeset.

The title page specifications are:

- Page size is 30 picas wide, 39 picas deep and top aligned.
- Title appears 9 picas down from top of page, is centered horizontally and is set in bold face 12 point type.
- Text dimensions on the title page are a 5 pica indent from the left margin, 13 picas down from the top of the page, 20 picas wide, 22 picas deep, justified horizontally and vertically.
- Page number appears in 12 point type, 3 picas below the last line of text, and is centered horizontally.

Specifications for Page 2 and all succeeding pages:

- Title appears 3 picas down from the top of the page, centered horizontally.
- Text dimensions are a 5 pica indent from the left margin, 6 picas down from the top of the page, 20 picas wide, 29 picas deep, justified horizontally and vertically.
- Text is set in 10 point type with a linespace of 12.
- Page number appears in 12 point type, 3 picas below the last line of text, and is centered horizontally.

Specifications can be set up in formats. Formats will contain compartments defining the positions of the header, text, and footer on each page. The footer format will contain a counter and dump counter command so the page number will appear sequentially on each page.

You Are Going To:

1. Create a format file containing formats for the title page, text, header and footer.
2. Name the format file "layout," the title format "title," the header format "top" and the footer format "bottom".
3. Create a multiple page document that automatically enters headers and footers.

Do The Following:

► Start a new file called "layout".

► Put the PowerView 10 in Edit mode:
[FILE FUNC] EDIT [EXECUTE]

► Enter the following format definitions for the job:
[PREC LC] [TYPE CMO] .DFMT,start [PREC LC] g<PAGE30,39,6,T><CMPT0,9,30,1,C,J>
<CMPT5,13,20,22,J,J><PS12<FT3<KO>Book Title [RETURN]
<PS10<LS12<FT1><FOOT::bottom::> [PREC UC] g [PREC UC] [TYPE CMX] [RETURN]
[PREC LC] [TYPE CMO] .DFMT,bottom [PREC LC] g<CMPT0,38,30,1,C,J><PS12><ADDC1,1>
[PREC LC] [TYPE CMO] .DMP,C1 [PREC UC] [TYPE CMX] <HEAD::top::> [PREC UC] g [PREC UC] [TYPE CMX] [RETURN]
[PREC LC] [TYPE CMO] .DFMT,top [PREC LC] g<CMPT0,3,30,1,C,J><CMPT5,6,20,29,J,J>
<PS12<FT3>Book Title [RETURN]
<PS10<FT1><FOOT::bottom::> [PREC UC] g [PREC UC] [TYPE CMX]

► Check the format syntax.

► [APND] [EXECUTE]

► [CLOSE FILE] [EXECUTE]

Explanation of formats

The title page format called "start" contains:

1. <**PAGE30,39,6,T**> This command sets up the page dimensions with a width of 30 picas, a depth of 39 picas, an interpage gap of 6 picas and top alignment.
2. <**CMPT0,9,30,1,C,J**> This compartment positions the book title on the title page.
3. <**CMPT5,13,20,22,J,J**> This compartment positions the text on the title page.
4. <**PS12**<**FT3**<**KO**> These are the parameters and copy for the header at the top of the page.
5. <**PS10**<**LS12**<**FT1**> These are the parameters for the text on the title page.
6. <**FOOT::bottom::**> This is the foot command. The PowerView 10 will automati-
   cally call it into the text when the compartment is full.

The footer format called "bottom" contains:

1. <**CMPT0,38,30,1,C,J**> This compartment positions the page number.
2. <**PS12**>This is the point size for the page number.
3. <**ADDC1,1**> This is the counter command that will begin the first page number at 1 and will add 1 each time the footer format "bottom" is called.
4. {**.DMP,C1**} This is the dump command that displays the counter (page number). A counter will not show in a job unless it is dumped.
5. <**HEAD::top::**> This is the head command which tells the PowerView 10 to call the header format "top" after the footer has been entered.

The header format called "top" contains:

1. <**CMPT0,3,30,1,C,J**> This compartment positions the book title on all succeeding pages.
2. <**CMPT5,6,20,29,J,J**> This compartment positions the text on all succeeding pages.
3. <**PS12**<**LS12**>**Book Title** These are the parameters and copy for the book title at the top of the page.

4. <**PS10**<**LS12**> These are the parameters for the text on the page.
5. <**FOOT::bottom::**> This is the foot command. The PowerView 10 will automatically call it into the text when the compartment is full.

You have created the format file for Exercise 1. Now create the text file.

► Start a new file called "simple".

► Load the format file "layout":
PREC LC TYPE CMO .LFD PREC LC g layout PREC UC g PREC UC TYPE CMX

► Call the format "start" after the loaded format file:
PREC LC TYPE CMO start PREC UC TYPE CMX

You are now ready to enter text. As you enter the text and reach the end of the defined page you will see a half-intensity footer format and the new page command <?NP> . The PowerView 10 will beep and display the "Page Overflow" message. When this happens, append the page. A half-intensity header format will appear at the top of the new file page and you can see the header text displayed in the view window.

► Enter the following text:

Research into virtually every aspect of typography has been in progress since the beginning of the present century. These experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research it's readily apparent that offices equipped with type-producing systems will enjoy substantial advantages over competitors restricted to typewriter and word processor style output.

Researchers have shown that setting text in type makes it easier to read and absorb than typewriter style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design qualities (character spacing, shape, density,

crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces. That's one of many reasons why typeset copy is more legible than typewriter style copy. Typographic design is an extraordinary blend of art and science that produces type with maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of turning out designs that are reliable, effective communication tools. That's why only a very small number of new typefaces is developed each year.

It's no wonder, then, that experiments by readability researchers demonstrate that type's custom fitting of each character in the alphabet enables typeset copy to be read more quickly and easily than typewriting. With type's precisely adjusted letterspacing, the eye instantly recognizes the shapes of entire words, instead of each individual letter. This makes for quicker and more effective communication of written messages. In today's fast-moving business world, where more and more information needs to be communicated in less and less time, type's superior readability gives the communicator an important "edge" that's more valuable than ever before.

When you reach the end of the job, manually enter the footer format. This will put the last page number on the bottom of the last page.

► Call the footer into the text to end the page:

PREC LC | TYPE CMO | bottom | PREC UC | TYPE CMX

► APND | EXECUTE | CLOSE FILE | EXECUTE

The text you see on your view window will look similar to the next two pages.

Book Title

Research into virtually every aspect of typography has been in progress since the beginning of the present century. These experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research, it's readily apparent that offices equipped with type-producing systems will enjoy substantial advantages over competitors restricted to typewriter and word processor style output.

Researchers have shown that setting text in type makes it easier to read and absorb than typewriter style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design qualities (character spacing, shape, density, crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces. That's one of many reasons why typeset copy is more legible than typewriter style copy. Typographic design is an extraordinary blend of art and science that produces type with maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of turning out designs that are reliable, effective commu-

nication tools. That's why only a very small number of new typefaces is developed each year.

It's no wonder, then, that experiments by readability researchers demonstrate that type's custom fitting of each character in the alphabet enables typeset copy to be read more quickly and easily than typewriting. With type's precisely adjusted letterspacing, the eye instantly recognizes the shape of entire words, instead of each individual letter. This makes for quicker and more effective communication of written messages. In today's fast-moving business world, where more and more information needs to be communicated in less and less time, type's superior readability gives the communicator an important "edge" that's more valuable than ever before.

Documents with Alternating Page Layouts

The following rules apply to documents with alternating page layouts:

1. The header on each left hand page is the same.
2. The footer on each left hand page is the same.
3. The header on each right hand page is the same.
4. The footer on each right hand page is the same.

The only difference between typesetting a simple document and a doument with alternating page layouts is the number of formats needed for the job. A simple document has a format for the header, one for the footer and one for the title page, if necessary. A document with alternating page layouts has a format for the left header, one for the left footer, one for the right header, one for the right footer and one for the title page, if necessary. The left header may be different than the right header; the left footer may be different than the right footer; but each left header will be the same and each right header will be the same. This is how copy flows in this type of a document:

- The left head calls the left foot.
- The left foot calls the right head.
- The right head calls the right foot.
- The right foot calls the left head and the cycle starts over again.

Save Command <SAVE>

The <SAVE> command is entered in a format or text file and stores, in PowerView 10 memory, the typesetting parameters immediately preceding it. These are the parameters that can be saved:

Point Size
Linespace
Font I.D.
Tab I.D. (<CT1>, <CT2>)
Pseudo Italic On
Pseudo Italic Off
Set Size

Only one set of parameters may be saved at a time. Previously saved parameters will be overwritten by an additional <SAVE> command and new

parameters.

This command eliminates having to re-enter the saved parameters when they are needed at a later time.

Restore Command <RSTR>

The <RSTR> command is entered where the saved parameters will be reactivated. This will happen automatically and will not appear on the screen; however, the status line on the screen will reflect the change. This command may be entered repeatedly and will restore the previously saved parameters each time.

Exercise 2—Documents with Alternate Page Layouts

Exercise 2 will show you how to set up the necessary formats to produce documents with alternating layouts. In Exercise 1, the header on each page is the book title, and the footer on each page is the page number. The headers and footers appear in the same place on each page.

In Exercise 2, you will create five new formats to do the following:

- Book title is centered on the first page.
- Author's name is quadded left at the top of each left hand page.
- Page number is quadded left at the bottom of each left hand page.
- Author's name is quadded right at the top of each right hand page.
- Page number is quadded right at the bottom of each right hand page.

You Are Going To:

1. Create a new format file called "bookfmt".
2. Define five new formats.
3. Create a new text file called "book".

Do The Following:

► Create a new format file called "bookfmt".

► Put your PowerView 10 in the Edit mode:
[FILE FUNC] EDIT [EXECUTE]

► Enter the following format definitions:
[PREC LC] [TYPE CMD] .DFMT,begin [PREC LC] g<PAGE30,30,4><CMPT0,5,20,1,C,J>
<PS12<FT3<KO>Book Title [RETURN]
<CMPT0,10,20,16,J,J><PS10<LS12<FT1><SAVE><FOOT::Rfoot::>
[PREC UC] g [PREC UC] [TYPE CMX] [RETURN]

[PREC LC] [TYPE CMD] .DFMT,Rfoot [PREC LC] g<CMPT0,28,20,1,R,J><PS10<FT3>
<ADDC1,1> [PREC LC] [TYPE CMD] .DMP,C1 [PREC UC] [TYPE CMX] <HEAD::Lhead::>
[PREC UC] g [PREC UC] [TYPE CMX] [RETURN]

PREC LC TYPE CMO .DFMT,Lhead PREC LC g<CMPT0,3,20,1,L,J><RSTR>
Author's Name RETURN
<CMPT0,5,20,21,J,J><FOOT::Lfoot::> PREC UC g PREC UC TYPE CMX RETURN

PREC LC TYPE CMO .DFMT,Lfoot PREC LC g<CMPT0,28,20,1,L,J><PS10<FT3>
<ADDC1,1> PREC LC TYPE CMO .DMP,C1 PREC UC TYPE CMX
<HEAD::Rhead::> PREC UC g PREC UC TYPE CMX RETURN

PREC LC TYPE CMO .DFMT,Rhead PREC LC g<CMPT0,3,20,1,R,J><RSTR>
Author's Name RETURN
<CMPT0,5,20,21,J,J><FOOT::Rfoot::> PREC UC g PREC UC TYPE CMX RETURN

Explanation of Formats:

The title page format "begin" contains:

1. **<PAGE30,30,4>** This command sets the width and depth of the page and space between pages.

2. **<CMPT0,5,20,1,C,J>** This compartment positions the book title.

3. **<PS12<FT3<KO>Book Title** These are the parameters and copy for the header at the top of the title page.

4. **<CMPT0,10,20,16,J,J>** This compartment positions text on the title page.

5. **<PS10<LS12<FT1>** These are the text parameters.

6. **<SAVE>** This is a new command that saves, in PowerView 10 memory, the
parameters immediately preceding it. In this case, it saves **<PS10<LS12<FT1>**.

7. **<Foot::Rfoot::>** This the foot command that tells the PowerView 10 to call the right footer.

The right foot format "Rfoot" contains:

1. <**CMPT0,28,20,1,R,J**> This compartment positions the right footer.

2. <**PS10**<**FT3**> These are the parameters for the footer.

3. <**ADDC1,1**> This is the counter command that sets the page number to increment sequentially.

4. {**.DMP,C1**} This is the command that displays the counter (page no.).

5. <**HEAD::Lhead::**> This is the head command that tells the PowerView 10 to call
the left header.

The left head format "Lhead" contains:

1. <**CMPT0,3,20,1,L,J**> This compartment positions the left header.

2. <**RSTR**> This is a new command. It restores the parameters previously saved with the <**SAVE**> command.

3. **Author's Name** This is the text for the header.

4. <**CMPT0,5,20,21,J,J**> This compartment positions the text on the page.

5. <**FOOT::Lfoot::**> This is the foot command that tells the PowerView 10 to call the
left footer.

The left foot command "Lfoot" contains:

1. <**CMPT0,28,20,1,L,J**> This compartment positions the left footer.

2. <**PS10**<**FT3**> These are the parameters for the footer.

3. <**ADDC1,1**> This is the counter command that sets the page number to increment sequentially.

4. {**.DMP,C1**} This is the command that displays the counter (page number).

5. <**HEAD::Rhead::**> This is the head command that tells the PowerView 10 to call the right header.

The right head command "Rhead" contains:

1. <**CMPT0,3,20,1,R,J**> This compartment positions the right header.

2. <**RSTR**> This command restores the parameters previously saved with the <**SAVE**> command.

3. **Author's Name** This is the text for the header.

4. <**CMPT0,5,20,21,J,J**> This compartment positions the text.

5. <**FOOT::Rfoot::**> This is the foot command that tells the PowerView 10 to call the right footer.

► Append the page and close the file.

You have defined formats that will output a document with alternating page layouts. The next step is to load the formats into a text file and output it.

► Start a new file called "book".
► Load the format file "bookfmt":
PREC LC TYPE CMO .LFD PREC LC g bookfmt PREC UC g PREC UC TYPE CMO
► Call the format "begin":
PREC LC TYPE CMO begin PREC UC TYPE CMX

► Enter the following text:

REMEMBER: Each time you the page breaks, you must append the file page.

Research into virtually every aspect of typography has been in progress since the beginning of the present century. These experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research, it's readily apparent that offices equipped with type-producing systems will enjoy substantial advantages over competitors restricted to typewriter and word processor style output.

Researchers have shown that setting text in type makes it easier to read and absorb than typewriter style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design qualities (character spacing, shape, density, crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces. That's one of many reasons why typeset copy is more legible than typewriter style copy. Typographic design is an extraordinary blend of art and science that produces type with maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of turning out designs that are reliable, effective communication tools. That's why only a very small number of new typefaces is developed each year.

It's no wonder, then, that experiments by readability researchers demonstrate that type's custom fitting of each character in the alphabet enables typeset copy to be read more quickly and easily than typewriting. With type's precisely adjusted letterspacing, the eye instantly recognizes the shape of entire words, instead of each individual letter. This makes for quicker and more effective communication of written messages. In today's fast-moving business world, where more and more information needs to be communicated in less and less time, type's superior readability gives the communicator an important "edge" that's more valuable than ever before.

Text continues on next page

Research into virtually every aspect of typography has been in progress since the beginning of the present century. These experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research, it's readily apparent that offices equipped with type-producing systems will enjoy substantial advantages over competitors restricted to typewriter and word processor style output.

Researchers have shown that setting text in type makes it easier to read and absorb than typewriter style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design qualities (character spacing, shape, density, crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces. That's one of many reasons why typeset copy is more legible than typewriter style copy. Typographic design is an extraordinary blend of art and science that produces type with maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of turning out designs that are reliable, effective communication tools. That's why only a very small number of new typefaces is developed each year.

It's no wonder, then, that experiments by readability researchers demonstrate that type's custom fitting of each character in the alphabet enables typeset copy to be read more quickly and easily than typewriting. With type's precisely adjusted letterspacing, the eye instantly recognizes the shape of entire words, instead of each individual letter. This makes for quicker and more effective communication of written messages. In today's fast-moving business world, where more and more information needs to be communicated in less and less time, type's superior readability gives the communicator an important "edge" that's more valuable than ever before.

Research into virtually every aspect of typography has been in progress since the beginning of the present century.

► Append the page and close the file.

The text you see in the view window will look similar to that shown below and on the following pages.

Book Title

Research into virtually every aspect of typography has been in progress since the beginning of the present century. These experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research, it's readily apparent that offices equipped with type-producing systems will enjoy substantial advantages over competitors restricted to typewriter and word processor style output.

Researchers have shown that setting text in type makes it easier to read and absorb than typewriter style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design qualities (character spacing, shape, density, crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces. That's one of

1

many reasons why typeset copy is more legible than typewriter style copy. Typographic design is an extraordinary blend of art and science that produces type with maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of turning out designs that are reliable, effective communication tools. That's why only a very small number of new typefaces is developed each year.

It's no wonder, then, that experiments by readability researchers demonstrate that type's custom fitting of each character in the alphabet enables typeset copy to be read more quickly and easily than typewriting. With type's precisely adjusted letterspacing, the eye instantly recognizes the shape of entire words, instead of each individual letter. This makes for quicker and more effective communication of written messages. In today's fast-moving business world, where more and more information needs to be communicated in less and less time, type's superior readability gives the communicator an important "edge" that's more valuable than ever before.

2

Research into virtually every aspect of typography has been in progress since the beginning of the present century. These experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research, it's readily apparent that offices equipped with type-producing systems will enjoy substantial advantages over competitors restricted to typewriter and word processor style output.

Researchers have shown that setting text in type makes it easier to read and absorb than typewriter style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design qualities (character spacing, shape, density, crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces. That's one of many reasons why typeset copy is more legible than typewriter style copy. Typographic design is an extraordinary blend of art and science that produces type with maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of

turning out designs that are reliable, effective communication tools. That's why only a very small number of new typefaces is developed each year.

It's no wonder, then, that experiments by readability researchers demonstrate that type's custom fitting of each character in the alphabet enables typeset copy to be read more quickly and easily than typewriting. With type's precisely adjusted letterspacing, the eye instantly recognizes the shape of entire words, instead of each individual letter. This makes for quicker and more effective communication of written messages. In today's fast-moving business world, where more and more information needs to be communicated in less and less time, type's superior readability gives the communicator an important "edge" that's more valuable than ever before.

Research into virtually every aspect of typography has been in progress since the beginning of the present century.

4

Setting Page Numbers That Do Not Start At No. 1

You may need to set a counter equal to a number other than 1.

For example, you have corrected a file containing a paperback book and decide to split the file to make typesetting easier. The book is 50 output pages so the second file will start on page 26.

To set page numbers that do not start at 1:

- ► Call the second file to the screen.
- ► Reload the format file and the starting format. This is necessary because the format file and starting format already loaded on page 1 will not carry forward after the file is split. Included are formats for alternating headers and footers like those in Exercise 2.
- ► Set counter C1 equal to 25:

<EQUC1,25>

The equal counter will set the page number at 25. It will not output as 25 because the footer format in the format file is set to increment by one digit each time it is called. Therefore, when the footer format is called and dumped, it will increment and output as 26.

Complex Documents

Format files for simple documents and documents with alternating page layouts are not too complicated. Basically, you set up formats to position any text and/or ruling in the headers and footers. Even though you have a compartment defined in the format file, once you begin typesetting the document, you may still enter X/Y coordinates and/or compartments to position text on the page.

Complex documents are those having a different header and footer on each page. The format file will contain a separate format for each header and footer in the document. This might work well for a short document, but it is not the recommended course for one that contains many pages.

File Management

An output page equals one file page. The PowerView 10 will tell you when to start a new page by displaying the "Page Overflow" message.

If you want to proofread a paged document before it goes to the typesetter, it is recommended that you place the PowerView 10 in edit mode. Make all changes in the document. When all corrections have been made, place the PowerView 10 in compose mode and repage the document.

When you open a multi-page, paged file, the PowerView 10 will add an extra half-intensity new page command at the bottom of each page. You must delete the second new page command and replace the page to scroll through the file.

APPLICATIONS

Introduction

This section includes nine sample applications for the Power Page Option. The formats used to achieve the results you will see are not the only methods available. Experiment with Power Page—you may find easier ways to complete these jobs. Studying these examples and adapting them to the applications you use frequently will enable you to use the full capabilties of the Power Page Option. Try them and experiment with your own ideas.

PUBLICATION

The example on the following page illustrates how Power Page can be used to simplify typesetting numbered paragraphs, tabs, and headings. The formats for this job include setting counters, dumping the counters, and defining all parameters and tabs within the formats. The use of formats allows the text file to be free of typesetting commands. In addition to entering the text in the job file, you will load the format file and call the various formats.

SPELLING

PREFERENCES OF SPECIAL GROUPS

1.1 British versus American spelling. The accepted practice is to change British spelling to American. This is to reduce the probability of inconsistencies.

1.2 Although common practice is to follow Webster's first-listed form for words with variant spellings, the variant may carry special connotations within certain disciplines, and these should be respected.

PLURALS

1.3 Proper names. Names of persons and other capitalized names form the plural in the usual way, by adding s or es.

1.4 Exceptions to the general rule have to be made when the ending would suggest a false pronunciation.

1.5 Italicized words. If names of newspapers, books, or other italicized names are used in the plural, the inflectional ending is set in roman type.

PUNCTUATION

TERMINAL PUNCTUATION

2.1 A period is used to indicate the end of a declarative or an imperative sentence.

2.2 A period should be omitted at the end of a sentence that is included within another sentence.

QUESTION MARK

2.3 The question mark, or interrogation point, is used to mark a query or to express an editorial doubt.

2.4 Indirect questions should not be followed by a question mark.

This is one suggested format file for this example:

{**.DFMT,begin(<LL12<LL25<KO<TB1,0,2,L<>)**}

{**.DFMT,head(<ALD14<PS14<FT3<ADDC1,1<EQUC2,1>)**}

{**.DFMT,sub(<ALD12<PS12<FT3>)**}

{**.DFMT,text(<ALD10<CT1<PS10<FT3<KX>{.DMP,C1}.{.DMP,C2}**
<KO<ADDC2,1<CT2<FT1<PS10>)}

You text file would follow this pattern:

{**.LFD(spellfmt)**}{**begin**}
{**head**}SPELLING←↵
{**sub**}PREFERENCES OF SPECIAL GROUPS←↵
{**text**}British versus American spelling. The accepted practice is to change British spelling to American. This is to reduce the probability of inconsistencies.←↵

You will call the format {**head**} before each major head, the format {**sub**} before each subhead, and the format {**text**} before each block of text. The numbers are automatically dumped and the parameters are changed with each format call.

For more practice, try the OUTLINE application on the next page.

OUTLINE

This application illustrates the ease of entering a standard outline and uses counters to dump and qualify numbers.

I. The Production Cycle
II. Calling the File
    A. Call Page, disk drive number, file name, Execute
    B. Opens the file, bringing copy to screen
III. Editing
    A. File must be open
    B. Arrow keys move cursor one character or continuously
    C. Scroll Up and Scroll Down—move copy through screen memory
    D. Limit key followed by Arrow or Scroll key—limit of movement
        1. Limit, Arrow Left or Right—cursor to end of screen line
        2. Limit, Arrow Up or Down—cursor to top or bottom of screen, then ten lines
        3. Limit, Scroll Up—cursor to bottom of job
        4. Limit, Scroll Down—cursor to top of job
IV. Screen Modes
    A. Composition mode
    B. Edit mode
V. Replacing the File
    A. Replace Page, Execute
    B. Close File, Execute

Your format file might look like this:

{.DFMT,layout(<PS10<LS12<FT1<LL30<KO<AL<TB1,0,1.6,R<TB2,2,28,L><TB3,3.6,26.6,L<TB4,5,25,L>)}

{.DFMT,sub1(<ADDC1,1<CT1>{.DMP,C1,R}.<CT2>)}

{.DFMT,sub2,A(<EQUC2,0<CT2>Ⓓ1.<CT3>)}

{.DFMT,sub3(<CT3<ADDC2,1>{.DMP,C2}<CT4>)}

The job file will follow this pattern:

{.LFD(outlinefmt)}{layout}
{sub1}The Production Cycle↵
{sub1}Calling the File↵
{sub2,A}Call Page, disk drive number, file name, Execute↵
{sub2,B}Opens the file, bringing copy to screen↵

Load the format file and call the format layout. This will set the tabs and parameters.

Call the format **{sub1}**before each roman numeral head. This will dump the correct roman numeral and call the correct tab.

Call the format **{sub2,Ⓓ1}** before each capital letter head. Fill the data slot with the correct letter.

Call the format **{sub3}** before each arabic head. This will dump the correct arabic numeral and call the correct tab.

Follow each entry by a tab return.

For more practice, try the SCHEDULE application on the next page.

SCHEDULE

The following application illustrates a typical schedule. It uses variables and escape characters.

CALENDAR

Sunday
Morning Worship—9:30. Coffee and brownies following the service.

Monday
Young People's Bible Class—7 p.m. Leader: Reverend Barney Browne.

Tuesday
Women's Group—6 p.m. Pot Luck Supper. Contact Mrs. Browne for ticket information. Child care is provided.

Wednesday
Young People's Group—7 p.m. at John Smith's house. Bring your ski trip pictures.

Thursday
Adult Bible Class—7 p.m. Leader: Rev. Browne

Friday
Young People's Group—6-10 p.m. Skating Party. Refreshments. Chaperones needed.

Saturday
Couple's Club—8 p.m. Bob Browne will show slides of Biscayne Bay. Refreshments will be served.

{.DFMT,start(<PS14<LS24<FT3<LL20<KO<AL>CALENDAR
<ALD48>)}

{.DFMT,head,day(<PS12<LS14<FT3>Ⓓ1←↵
)}

{.DFMT,text,topic(<PS10<LS12<FT1>Ⓓ1←↵
)}

The text file will follow this pattern:

{.LFD(calfmt)}{start}
{head,Sunday}{text,Morning Worship—9:30. Coffee and brownies following the service.}
{head,Monday}{text,Young People's Bible Class—7 p.m. Leader: Reverend Barney Browne}

Load the format file and call the format {**start**} to enter the title and parameters.

Enter the format {**head,**} followed by the day of the week. This changes point size and font, quads the word left, and returns the carriage to the left margin.

Enter the format {**text,**} followed by the event, time and place. This changes point size and font, and quads the last line left.

Try the PRICE LIST application on the next page.

PRICE LIST

This is a typical price list, using formats and variables.

DETAILED INDEX BY SERIES STYLE NUMBER

Part No.	Item Description	Page
02-1000	Broadloom Carpet	332
04-1000	Oriental Design Rugs	332
04-2000	Accent Rugs	331
04-4000	Handwoven Oriental Rugs	330
04-5000	Braided Rugs	331
05-4000	Ellen Adams Bedding	324
09-1000	Table Lamps	345
09-3000	Chandeliers and Wall Lamps	367
11-5000	Georgian Bedrooms	289
12-3000	Decorative Accents and Pieces	456
13-4000	Classic Bedrooms	590
21-7000	Sleeper Sofas	625
31-3000	Occasional Tables	629
41-2000	Wall Clocks	645
65-4000	Glass Sculptures	675

Your format file will look similiar to this:

{.DFMT,start(<PS16<LS12<FT3<LL30<KO>DETAILED INDEX BY SERIES STYLE NUMBER$^{E}_{L}$
<PS12<ALD4>$^{I\ B\ E}_{CR\ \ L}$
<ALD18<PS10<TB1,0,5,L<TB2,6,24,J<CT1>Part No.<CT2>Item Description <IS>Page^{T}↵
)}

{.DFMT,text,number,name,page(<KX<FT3<CT1>Ⓓ1<CT2>Ⓓ2<IS>^{L}.Ⓓ3^{T}↵
)}

The text file will follow this pattern:

{.LFD(pricelistfmt)}{start}
{text,02-1000,Broadloom Carpet,332}
{text,04-1000,Oriental Design Rugs,332}
{text,04-4000,Accent Rugs,331}

Load the format file and call the format {**start**}. This will set the parameters, tabs and headings.

Call the format {**text,,,**} followed by the part number, name and page number of each entry. This changes the parameter and sets the tabs.

For more practice, try the DISPLAY AD on the next page.

DISPLAY AD

This display ad uses variables, tabs in formats and text in formats.

BAR-B-QUE SPECIALS		
Extra Lean Ground Sirloin	5 lb. bag	$1.79
Hamburger Patties	5 lb. bag	$1.89
Boneless Leg of Lamb	avg. wt. 4 lbs.	$3.21
Extra Lean Skirt Steak	5 lb. pkg.	$2.46
Prime Sirloin Tips	5 lb. pkg.	$2.48
Boneless Sirloin Strip	avg. wt. 10 oz.	$3.89
Homemade Shishkabob	on skewers	$2.99

Your format file will look similar to this:

{.DFMT,layout(<KO<LL24<PS24<LS24<FT1>E L
I B C L ←↵
BAR-B-QUE SPECIALS<ALD6>E L
I B C L ←↵
<TB1,0,10,L<TB2,11,6,C<TB3,18,6,R>)}

{.DFMT,meat,name(<PS18<LS18<CT1>Ⓓ1)}

{.DFMT,weight,amount(<PS10<CT2>Ⓓ1 T↵
)}
{.DFMT,price,amount(<PS40<FT3<CT3>Ⓓ1C R
<ALD12<PS24<FT1>I B C L <LS24>←↵
)}

The text file will follow this pattern:

{.LFD(grocfmt)}{layout}
{meat,Extra Lean}{weight,5 lb. bag}{meat,Ground Sirloin}{price,$1.79}
{meat,Hamburger}{weight,5 lb. bag}{meat,Patties}{price,$1.89}

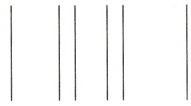
Load the format file and call the format {**layout**}. This enters the parameters and sets the heading.

Call the format {**meat**} followed by the type of meat. Call the format {**weight**} followed by the weight of the meat. Call the format {**meat**} again, followed by the second line of the type of meat. Call the format {**price**} followed by the price of the meat. The size and font changes and the horizontal rules will be automatically entered.

To practice area composition, try the RULED FORM application on the next page.

RULED FORM

This example uses the page command, XY positioning, the repeat function, boxes and tabs. It is set in a text file, not a format file.



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
First Class Mail Permit No. 44301 Arlington, MA

Postage will be paid by addressee

ATLANTIS
Subscription Processing Center
Box 1857
Greenwich, Connecticut 06830

<PAGE30,40,6><KO<PS10<LS12<XY9.6,0<DRV25,36,><XY11,0<DRV25,36,>
<XY11.6,0<DRV25,36<XY12.6,0<DRV25,36,<XY13,0<DRV25,36,>
<XY15,0<DRV25,36,<XY22,1.6<DBX25,84,,8><ALD20>
NO POSTAGE↔
NECESSARY↔
IF MAILED↔
IN THE↔
UNITED STATES↔

<XY22,9.6>{.RPT,14(<DRH100,,8><LS10>←
)}
<LS12<XY0,8.6<DBX25,42,,20<ALD18<TB1,0,20,C<FT3<CT1>
BUSINESS REPLY CARD←
<FT1>First Class Mail Permit No. 44301 Arlington, MA←
☐ Postage will be paid by addressee←
←
←
<PS30<FT3>ATLANTIS←
<FT1<PS10>Subscription Processing Center←
Box 1857←
Greenwich, Connecticut 06830←

Try another area composition application on the next page.

DISPLAY AD

This display ad uses a page command, XY positioning and compartments. The entire ad is set in a text file; there is no format file.

October Clearance!

Famous Maker Corduroy

JACKETS

79.99

SAVE 50-80%

Lined Wool Blend

SKIRTSUITS

49.99

Here's just a sampling of the incredible purchases and incredible reductions you will find storewide! Incredible first day sale starts Friday, October 8th at 8 a.m. Incredible, or what? I mean, we're talking incredible. Some items actually encroach upon the limits of incredulity. So hurry down and take advantage of these incredible deals and incredible values at incredible prices. It's just plain incredible.

Foster's

An incredible name in incredible clothes
110 Peachtree Street NE • Atlanta • 555-1222

<PAGE30,51,2><PS24<LS12<FT11<KO<ACO<ACS2>
October Clearance!↔

<XY15,3<PS30<LS30<FT3>SAVE↔

50-80%↔

<XY0,1<DBX100,120,0,14<XY15,6.6<DBX10,120,0,14>
<CMPT0,1,14,10,c,c><PS12<LS24<FT1>Famous Maker Corduroy←

<PS18<LS60<FT3>JACKETS←

<PS60>79^{99}←

<CMPT15,6.6,15,10,C,C><PS12<LS24<FT1>Lined Wool Blend←

<PS18<LS60<FT3>SKIRTSUITS←

<PS60>49^{99}←

<CMPT10,12.6,14,4.8,J,J><CMPT0,,30,4,J,J><CMPT0,22.6,30,7,C,D>
<PS12<FT7<LS14>Here's just a sampling of the incredible purchases and incredible reductions you will find storewide! Incredible first day sale starts Friday, October 8th at 8 a.m. Incredible, or what? I mean we're talking incredible. Some items actually encroach upon the limits of incredulity. So hurry down and take advantage of these incredible deals and incredible values at incredible prices. It's just plain incredible.<FB>←

<DRH100>
<XY0,3.6<PS36<FT11<LS14>Foster's←

<PS10<LS12<FT1>An incredible name in incredible clothes←

110 Peachtree Street NE - Atlanta - 555-1212←

<DRH100>

Try the next application: Business Cards.

BUSINESS CARDS

The business card layout on the following page uses variables and compartments. Crop marks are included in the format. All you need to set up is one format file with a variable. When the format is called into the text file, you have to enter the name change for each different business card.

Skyway Airlines

Kathy Davis
Flight Coordinator

627 Statler Office Building
20 Providence Street
Boston, MA 02116
Phone (617) 482-1454

Skyway Airlines

David Pollar
Ground Crew Supervisor

627 Statler Office Building
20 Providence Street
Boston, MA 02116
Phone (617) 342-1233

Skyway Airlines

Steve Williams
Finance Administrator Manager
Business Systems Division

627 Statler Office Building
20 Providence Street
Boston, MA 02116
Phone (617) 482-1443

Skyway Airlines

Wendy Baldwin
Group Leader

627 Statler Office Building
20 Providence Street
Boston, MA 02116
Phone (617) 482-1433

The format file will look similar to this.

```
{.DFMT,start(<PAGE25,55,3,T><KO><ACO><ACS2><SRP50,18>EL
{crops}<CMPT3.6,3.6,18,9,C,><CMPT,15.6,,,C,><CMPT,27.6,,,C,>
<CMPT,39.6,,,C,><CMPT4,51,16,1,L,C>)}↵

↵
{.DFMT,crops(<XY2,0><DRV<XY23,0><DRV><XY,2><DRH,,1.6<XY23.6,2>
<DRH,,1.6>EL
<XY,14><DRH,,1.6<XY23.6,14><DRH,,1.6>EL
<XY,26><DRH,,1.6<XY23.6,26><DRH,,1.6>EL
<XY,38><DRH,,1.6<XY23.6,38><DRH,,1.6>EL
<XY,50><DRH,,1.6<XY23.6,50><DRH,,1.6<XY2,50.6><DRV<XY23,50.6>
<DRV>)}↵

↵
{.DFMT,card,name,title,phone(<PS14<LS10<FT11>Skyway Airlines←↵
<VIS><PS10<FT3>Ⓓ1↔↵
<PS8<FT1>Ⓓ 2↔↵
<VIS>627 Statler Office Building←↵
20 Providence Street←↵
Boston, MA 02116←↵
Phone (617) Ⓓ 3←<FB>)}↵

↵
{.DFMT,tag,date,cust. no.(<PS6>Date:: Ⓓ1□□Customer No.:: Ⓓ 2)}
```

The job file will look similar to this.

{.LFD(buscardfmt)}{start}{card,Kathy Davis,Flight Coordinator,482-1454}{card,David Pollar,Ground Crew Supervisor,342-1233}{card,Steve Williams,Finance Administrator Manager↔
Business Systems Division,482-1443}{card,Wendy Baldwin,Group Leader,482-1433}
{tag,6/2/84,2343}

If the same name, title, and phone number were to be entered four times the repeat function could be used. An example text file using the repeat function would look something like this:

{.LFD(buscardfmt)}{start}
{.RPT,4({card,Kathy Davis,Flight Coordinator,482-1454})}
{tag,6/2/84,2344}

Proceed to the Magazine Page layout that follows.

MAGAZINE PAGE LAYOUT

The format file for the magazine page layout on the following page includes a box command in a format, XY positioning, the page setup and compartments. Notice how the text flows around the box when it reaches that point in the various compartments.

The Sonoran Desert of North America is one of the most inhospitable places on Earth. By early July, two months have passed without rain, and during the day temperatures regularly climb above 100 degrees Fahrenheit. Many of the desert's plants and animals are dormant during this intensely dry period, waiting for the violent summer storms, locally called *chubacos*, that will revive them.

Of the many species that survive in the harsh Sonoran Desert, perhaps the most specialized is a toad known as Couch's Spadefoot. No ordinary amphibian, Couch's toad stays underground 10 to 11 months out of the year and takes maximum advantage of the one month that the desert is a suitable place for it to live. Within minutes after the July rains begin, tens of thousands of toads come bursting out of the ground to feed and reproduce. Then, almost as suddenly, they disappear until the next year.

The rains that cue Couch's Spadefoot must be sufficient to create pools large enough, about a foot deep, for breeding and larval development to occur. In any given pool, mating and egg depositing usually take place during only one night.

The toad uses an internal clock to emerge at the right time. As the summer rainy season approaches, the toads move closer to the surface. On nights of light rains, some can be found with only their eyes and the tops of their heads poked above ground. In addition, Couch's Spadefoot is extremely sensitive to all ground vibrations, like those that might be detected from pounding rains. Mark Dimmitt of the Arizona-Sonora Desert Museum has lured Couch's Spadefoot into complete emergence by simulating ground vibrations with an electric motor.

As soon as they emerge, the male toads gather at the pools and begin chorusing. The females, however, often take time to forage for insects before moving to the breeding pools. During the two to four weeks when the toad is nocturnally active, it must eat enough to last at least 11 months. To accomplish this, Couch's Spadefoot has evolved a stomach capable of storing over half its body weight in food. The stomachs of other toad species can hold, at best, little more than a third of their weight.

But eating is second priority for males who have converged at the pool actively seeking a female mate. Unlike many other toads that simply stand and call, these amorous males also cruise silently about the pool in search of a partner.

Mating and egg laying lasts the entire night. The breeding pair moves about the pool and the female attaches 100 to 200 eggs at a time to emergent vegetation while the male fertilizes the eggs. A large female lays up to 1,000 eggs in one evening, but by morning all the toads have left the pools and have dug into the ground to escape the heat. Only the eggs remain.

Now, as the hot desert sun begins to dry up the small pools, the race is on. With the fastest developmental rate known among Amphibia—nine days from egg to toadlet under ideal conditions—Couch's Spadefoot responds like a highly trained athlete. But often even this is not quick enough and the tadpoles dry up.

Rates of tadpole development are determined by the water temperature and nutrients in the pool. The rains bring with them bacteria and other organic matter that are consumed by the rapidly developing tadpoles. In water that approaches 100 degrees Fahrenheit, microorganisms grow very rapidly and algae, another important food source, soon blooms. In addition, Spadefoot tadpoles will eat anything they can find in their pools, including insects and even smaller tadpoles of other species.

The Sonoran Desert of North America is one of the most inhospitable places on Earth. By early July, two months have passed without rain, and during the day temperatures regularly climb above 100 degrees Fahrenheit. Many of the desert's plants and animals are dormant during this intensely dry period, waiting for the violent summer storms, locally called *chubacos*, that will revive them.

Of the many species that survive in the harsh Sonoran Desert, perhaps the most specialized is a toad known as Couch's Spadefoot. No ordinary amphibian, Couch's toad stays underground 10 to 11 months out of the year and takes maximum advantage of the one month that the desert is a suitable place for it to live. Within minutes after the July rains begin, tens of thousands of toads come bursting out of the ground to feed and reproduce. Then, almost as suddenly, they disappear until the next year.

The rains that cue Couch's Spadefoot must be sufficient to create pools large enough, about a foot deep, for breeding and larval development to occur. In any given pool, mating and egg depositing usually take place during only one night.

The toad uses an internal clock to emerge at the right time. As the summer rainy season approaches, the toads move closer to the surface. On nights of light rains, some can be found with only their eyes and the tops of their heads poked above ground. In addition, Couch's Spadefoot is sensitive to all ground vibrations, like those that might be detected from pounding rains.

"The rains that cue Couch's Spadefoot must be sufficient to create pools large enough ... for breeding and larval development to occur."

The format file for the magazine page layout will appear similar to this one.

{.DFMT,start,title(<PAGE36,53,4,T><XY0,0><DRH50><VBS0,6>↵
<PS12<FT3<ALD12>Ⓓ1←↵
<ALD3<DRH50>↵
<XY12.6,25><DBX50,72>↵
<XY12.6,27><IB1<IO><PS12<LS14<FT43>)}↵
↵
{.DFMT,text(<IX><PS9<LS11<FT41<CMPT0,3,11,,,><CMPT12.6,3,,21,,>
<CMPT,32,,,,><CMPT25,3,,21,,><CMPT,32,,,,>)}↵
↵

The job file will appear as follows. Note that the entire page of text is not entered; only the text within the box and the first paragraph are entered to illustrate how the text file will look.

{.LFD(magazinefmt)}{start,Survival}"The rains that cue Couch's Spadefoot must be
sufficient to create pools large enough ... for breeding and larval development to occur. "←↵
{text}The Sonoran Desert of North America is one of the most inhospitable places on Earth. By early July, two months have passed without rain, and during the day temperatures regularly climb above 100 degrees Fahrenheit. Many of the desert's plants and animals are dormant during this intensely dry period, waiting for the violent summer storms, locally called <**FT42**>chubacos<**FT41**>, that will revive them.←↵

For more practice, try the multiple-page text application on the next page.

MULTIPLE PAGE DOCUMENTS

The Multiple Page application on the following pages is divided into three parts. The first section illustrates a one-column layout, the second illustrates a two-column layout, and the third illustrates a three-column layout. The same text is used for each layout.

The various Power Page functions included in each layout are the page definition, compartments, headers, footers, counters, dumping counters, and vertical rules and tabs within formats. The same basic design is used for each layout; the only changes are those in compartment sizes, the placement of vertical rules and tab parameters. In addition, the format name called into the text file will change with each layout.

Remember, each PowerView 10 file page can contain only one output page. When you fill an output page you must append it and begin a new file page.

The Use of Typography in Office Communication

Information Overload. The contemporary business community finds itself beset by a superabundance of paper-based information. Myriad proposals, plans, memos, letters, contracts, reports, bulletins, and other printed communications cross each businessperson's desk each day, and this deluge is unlikely to recede in the foreseeable future.

It has been estimated that more than 72 billion pieces of written business communication are being produced yearly. All of these documents, of course, need to be read, sorted, distributed, output, stored, and/or retrieved. Computerized office automation, which was heralded as the cure for business's communication headaches, is in fact creating *more* paper. That today's office is far from paperless is more readily evidenced by the explosive growth of overnight document delivery services such as Airborne and Federal Express. Similarly, corporate purchasing departments report that they are buying more printer and photocopier paper today than ever before.

It has become evident that office communication simply cannot be productive in totally electronic form. A computer screen lacks the special "feel" and profitability of paper-based materials. Increasingly, businesspeople are finding that video display terminals are often impractical vehicles for everyday office communications.

Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data:

1) employees are unable to process all the information they receive
2) employees tend to disregard some information altogether
3) employees often divert information to inappropriate receivers
4) employees arbitrarily deal with information without regard to its relative importance
5) employees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences
6) employees attempt to "pigeonhole" information according to their own arbitrary categories

The Typographic Solution. Up to now, the technological advances responsible for automating office work have only aggravated information overload and its associated problems. Powerful, ergonomically engineered computers and word processors facilitate the production of more and more information at ever-accelerating speeds and leave the beleaguered office employee more overburdened with paper than ever before. The computer industry has developed increasingly productive input systems, while neglecting to address the productivity of those systems' output capabilities.

Today, however, new breakthroughs in output technologies promise to complete the office automation revolution by making printed information easier to read, digest, and remember.

The end-product of these newly developed output technologies is literally right before your eyes. Typographic communication like that you are now reading ensures that business messages will reach more people more quickly, that it will be read and understood more easily, and that they will carry far more credibility than ordinary typewritten communication. Typographic office communication offers the following benefits:

- more readable communication
- more economical communication resulting from reduced paper, paper storage, postage and dling costs
- more memorable communication
- more appealing communication
- more professional-looking communication
- more creative communication through the use of eye-catching layouts and graphics

Typographic composition itself is, of course, by no means new. It has been used in the western world since Gutenberg's invention of movable type more than 500 years ago. For centuries mass communicators, such as professional printers, publishers, and advertising people, have truly exploited type's matchless communicative power to produce materials that are attention-getting, attractive, and extraordinarily easy to absorb and retain. Now, at precisely the time when it is most needed by the information-saturated office environment, type is available for use in business offices of every size and description.

The Final Phase of Office Automation. Creating professional-looking typesetting is no longer an expensive, time-consuming process. In fact newly developed typographic compositions systems enable anyone who can use a standard typewriter keyboard to turn out professional-quality documents quickly and conveniently.

These systems consist of a desktop personal computer or word processor connected to a small output unit. Operating them is no more difficult than typewriting: there are no complicated keyboard skills to master and output can be on the same kind of paper used for ordinary typewriting.

The new typographic composition systems represent the next logical step in office automation. Office system development has traditionally been focused on improving input speed and efficiency with simplified keyboarding, easy-to-use software, etc. Output, on the other hand, has remained no better than that which has always been available from a conventional office typewriter.

Today, users of electronic office equipment can enjoy high-quality output that matches the capabilities of their sophisticated input systems. The missing link in office automation is missing no longer.

An Efficient Business Communication Medium. Research into virtually every aspect of typography has been in progress since the beginning of the present century. These carefully controlled experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research it is readily apparent that business offices

The Use of Typography in Office Communication

Information Overload. The contemporary business community finds itself beset by a superabundance of paper-based information. Myriad proposals, plans, memos, letters, contracts, reports, bulletins, and other printed communications cross each businessperson's desk each day, and this deluge is unlikely to recede in the foreseeable future.

It has been estimated that more than 72 billion pieces of written business communication are being produced yearly. All of these documents, of course, need to be read, sorted, distributed, output, stored, and/or retrieved. Computerized office automation, which was heralded as the cure for business's communication headaches, is in fact creating *more* paper. That today's office is far from paperless is more readily evidenced by the explosive growth of overnight document delivery services such as Airborne and Federal Express. Similarly, corporate purchasing departments report that they are buying more printer and photocopier paper today than ever before.

It has become evident that office communication simply cannot be productive in totally electronic form. A computer screen lacks the special "feel" and profitability of paper-based materials. Increasingly, businesspeople are finding that video display terminals are often impractical vehicles for everyday office communications.

Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data:

1) employees are unable to process all the information they receive
2) employees tend to disregard some information altogether
3) employees often divert information to inappropriate receivers
4) employees arbitrarily deal with information without regard to its importance
5) employees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences
6) employees attempt to "pigeonhole" information according to their own arbitrary categories

The Typographic Solution. Up to now, the technological advances responsible for automating office work have only aggravated information overload and its associated problems. Powerful, ergonomically engineered computers and word processors facilitate the production of more and more information at ever-accelerating speeds and leave the beleaguered office employee more overburdened with paper than ever before. The computer industry has developed increasingly productive input systems, while neglecting to address the productivity of those systems' output capabilities.

Today, however, new breakthroughs in output technologies promise to complete the office automation revolution by making printed information easier to read, digest, and remember.

The end-product of these newly developed output technologies is literally right before your eyes. Typographic communication like that you are now reading ensures that business messages will reach more people more quickly, that it will be read and understood more easily, and that they will carry far more credibility than ordinary typewritten communication. Typographic office communication offers the following benefits:

- more readable communication
- more economical communication resulting from reduced paper, paper storage, postage and handling costs
- more memorable communication
- more appealing communication
- more professional-looking communication
- more creative communication through the use of eye-catching layouts and graphics

Typographic composition itself is, of course, by no means new. It has been used in the western world since Gutenberg's invention of movable type more than 500 years ago. For centuries mass communicators, such as professional printers, publishers, and advertising people, have truly exploited type's matchless communicative power to produce materials that are attention-getting, attractive, and extraordinarily easy to absorb and retain. Now, at precisely the time when it is most needed by the information-saturated office environment, type is available for use in business offices of every size and description.

The Final Phase of Office Automation. Creating professional-looking typesetting is no longer an expensive, time-consuming process. In fact newly developed typographic compositions systems enable anyone who can use a standard typewriter keyboard to turn out professional-quality documents quickly and conveniently.

These systems consist of a desktop personal computer or word processor connected to a small output unit. Operating them is no more difficult than typewriting: there are no complicated keyboard skills to master and output can be on the same kind of paper used for ordinary typewriting.

The new typographic composition systems represent the next logical step in office automation. Office system development has traditionally been focused on improving input speed and efficiency with simplified keyboarding, easy-to-use software, etc. Output, on the other hand, has remained no better than that which has always been available from a conventional office typewriter.

Today, users of electronic office equipment can enjoy high-quality output that matches the capabilities of their sophisticated input systems. The missing link in office automation is missing no longer.

An Efficient Business Communication Medium. Research into virtually every aspect of typography has been in progress since the beginning of the present century. These carefully controlled experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research it is readily apparent that business offices equipped with typographic composition systems will enjoy substantial advantages over competitors restricted to typewriter and word processor-style output.

Comprehension and Speed. Researchers have shown that setting text in type makes it easier to

The Use of Typography in Office Communication

Information Overload. The contemporary business community finds itself beset by a superabundance of paper-based information. Myriad proposals, plans, memos, letters, contracts, reports, bulletins, and other printed communications cross each businessperson's desk each day, and this deluge is unlikely to recede in the foreseeable future.

It has been estimated that more than 72 billion pieces of written business communication are being produced yearly. All of these documents, of course, need to be read, sorted, distributed, output, stored, and/or retrieved. Computerized office automation, which was heralded as the cure for business's communication headaches, is in fact creating *more* paper. That today's office is far from paperless is more readily evidenced by the explosive growth of overnight document delivery services such as Airborne and Federal Express. Similarly, corporate purchasing departments report that they are buying more printer and photocopier paper today than ever before.

It has become evident that office communication simply cannot be productive in totally electronic form. A computer screen lacks the special "feel" and profitability of paper-based materials. Increasingly, businesspeople are finding that video display terminals are often impractical vehicles for everyday office communications.

Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data:

1) employees are unable to process all the information they receive
2) employees tend to disregard some information altogether
3) employees often divert information to inappropriate receivers
4) employees arbitrarily deal with information without regard to its relative importance
5) employees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences
6) employees attempt to "pigeonhole" information according to their own arbitrary categories

The Typographic Solution. Up to now, the technological advances responsible for automating office work have only aggravated information overload and its associated problems. Powerful, ergonomically engineered computers and word processors facilitate the production

of more and more information at ever-accelerating speeds and leave the beleaguered office employee more overburdened with paper than ever before. The computer industry has developed increasingly productive input systems, while neglecting to address the productivity of those systems' output capabilities.

Today, however, new breakthroughs in output technologies promise to complete the office automation revolution by making printed information easier to read, digest, and remember.

The end-product of these newly developed output technologies is literally right before your eyes. Typographic communication like that you are now reading ensures that business messages will reach more people more quickly, that it will be read and understood more easily, and that they will carry far more credibility than ordinary typewritten communication. Typographic office communication offers the following benefits:

- more readable communication
- more economical communication resulting from reduced paper, paper storage, postage and handling costs
- more memorable communication
- more appealing communication
- more professional-looking communication
- more creative communication

Typographic composition itself is, of course, by no means new. It has been used in the western world since Gutenberg's invention of movable type more than 500 years ago. For centuries mass communicators, such as professional printers, publishers, and advertising people, have truly exploited type's matchless communicative power to produce materials that are attention-getting, attractive, and extraordinarily easy to absorb and retain. Now, at precisely the time when it is most needed by the information-saturated office environment, type is available for use in business offices of every size and description.

The Final Phase of Office Automation. Creating professional-looking typesetting is no longer an expensive, time-consuming process. In fact newly developed typographic compositions systems enable anyone who can use a standard typewriter keyboard to turn out professional-quality documents quickly and conveniently.

These systems consist of a desktop personal computer or word processor connected to a small output unit. Operating them is no more difficult than typewriting: there are no complicated keyboard skills to master and output can be on the same kind of paper used for ordinary typewriting.

The new typographic composition systems represent the next logical step in office automation. Office system development has traditionally been focused on improving input speed and efficiency with simplified keyboarding, easy-to-use software, etc. Output, on the other hand, has remained no better than that which has always been available from a conventional office typewriter.

Today, users of electronic office equipment can enjoy high-quality output that matches the capabilities of their sophisticated input systems. The missing link in office automation is missing no longer.

An Efficient Business Communication Medium. Research into virtually every aspect of typography has been in progress since the beginning of the present century. These carefully controlled experiments in psychology, linguistics, education, and other fields have consistently demonstrated that type is the most efficient way to communicate on paper. From this research it is readily apparent that business offices equipped with typographic composition systems will enjoy substantial advantages over competitors restricted to typewriter and word processor-style output.

Comprehension and Speed. Researchers have shown that setting text in type makes it easier to read and absorb than typewriter-style output. This ergonomic or "reader-friendly" quality is due largely to the fact that professionally developed type has built-in design attributes (character spacing, shape, density, crispness, etc.) that cannot be equalled by the limited selection of standard typewriter faces.

Typographic design is an extraordinary blend of art and science that produces type with both maximum eye-appeal and maximum readability. Only a handful of people in the world today are capable of turning out type designs that are reliable, effective communications tools, and only a very small number of new typefaces are developed each year.

Experiments have demonstrated that an important factor in typeset

One format file is used and includes formats for the three different layouts. The format file will look similiar to this:

```
{.DFMT,start(<PAGE45,60,6>{box}<EQUC1,1<SB6,4,12<KO<ACO<ACS2>
<CRD36>)}
{.DFMT,chaphead(<CMPT5,5,35,20,C,C><FT903<PS36<LS36>The Use of
Typography in
Office Communication)}
{.DFMT,tab(<TB1,1,1,L><TB2,2,{.DMP,C2},L>)}
{.DFMT,layout1(<EQUC2,33><CMPT5,27,35,25,J,J><PS10<LS12<FT7>
<FOOT::rightf1::>)}
{.DFMT,rightf1(CR
<CMPT5,53,35,1,R,D><PS12<FT8>{.DMP,C1}<ADDC1,1>
<HEAD::lefth1::>)}
{.DFMT,lefth1(CR
{why}<CMPT5,8,35,44,J,J><PS10<LS12<FT7><FOOT::leftf1::>EL
)}
{.DFMT,leftf1(CR
<CMPT5,53,35,1,L,D><PS12<FT8>{.DMP,C1}<ADDC1,1>)}
```

(Continued)

```
{.DFMT,layout2(<EQUC2,14><CMPT5,27,16.6,25,J,J<CMPT23.6,27,,,,>
<PS10<LS12<FT7><FOOT::rightf2::>)}
{.DFMT,rightf2(CR
<CMPT5,53,35,1,R,D><PS12<FT8>{.DMP,C1}<ADDC1,1><HEAD::lefth2::>)}
{.DFMT,lefth2(CR
{why}<DRV50,560,18.6><CMPT5,8,16.6,44,J,J><CMPT23.6,8,,,,>
<PS10><LS12<FT7><FOOT::leftf2::>EL
)}
{.DFMT,leftf2(CR
<CMPT5,53,35,1,L,D><PS12<FT8>{.DMP,C1}<ADDC1,1>)}
{.DFMT,layout3(<EQUC2,8><CMPT5,27,10.6,25,J,J<CMPT17.3,27,,,,>
<CMPT29.6,27,,,I,><PS9<LS11<FT7><FOOT::rightf3::>)}
{.DFMT,rightf3(CR
<CMPT5,53,35,1,R,D><PS12<FT8>{.DMP,C1}<ADDC1,1><HEAD::lefth3::>)}
{.DFMT,lefth3(CR
{why}<XY16.4,5.4><DRV50,560<DRV50,560,12.3><CMPT5,8,10.6,44,I,j>
<CMPT17.3,8,,,,<CMPT29.6,8,,,,<PS9<LS11<FT7><FOOT::leftf3::>EL
)}
{.DFMT,leftf3(CR
<CMPT5,53,35,1,L,D><PS12<FT8>{.DMP,C1}<ADDC1,1>)}
{.DFMT,box(<XY4,4><DBX50,624,0,37>)}EL
{.DFMT,why({box}<XY4,5><PS12<FT903<LL37>Why Type?↔EL
<ALD4<DRH50>)}EL
```

The job files will appear as follows. Notice that the only change in the format calls is the name of the format "layout," which changes to "layout1," "layout2" or "layout3," depending on your choice of a one-column, two-column or three-column layout. The format name you enter will call out the proper nested formats that are all defined in the format file.

The job file for a one-column layout will look similar to the following file.

{**.LFD(bookfmt)**}{**start**}{**chaphead**}{**layout1**}<**FT9**>Information Overload.
<**FT7**>The contemporary business community finds itself beset by a super-abundance of paper-based information. Myriad proposals, plans, memos, letters, contracts, reports, bulletins, and other printed communications cross each business person's desk each day, and this deluge is unlikely to recede in the foreseeable future.←
☐It has been estimated that more than 72 billion pieces of written business communication are being produced yearly. All of these documents, of course, need to be read, sorted, distributed, output, stored, and/or retrieved. Computerized office automation, which was heralded as the cure for business's communication headaches, is in fact creating <**FT8**>*more* <**FT7**>paper. That today's office is far from paperless is more readily evidenced by the explosive growth of overnight document delivery services such as Airborne and Federal Express. Similarly, corporate purchasing departments report that they are buying more printer and photocopier paper today than ever before.←
☐It has become evident that office communication simply cannot be productive in totally electronic form. A computer screen lacks the special "feel" and profitability of paper-based materials. Increasingly, businesspeople are finding that video display terminals are often impractical vehicles for everyday office communications.←
☐Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data:←
{**tab**}1)employees are unable to process all the information they receive
2)employees tend to disregard some information altogether
3)employees often divert information to inappropriate receivers

(Text continues on next page)

T4)Temployees arbitrarily deal with information without regard to its relative importance

T5)Temployees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences

T6)Temployees attempt to "pigeonhole" information according to their own arbitrary categories

<**FT9**>The Typographic Solution.<**FT7**>Up to now, the technological advances responsible for automating office work . . . .

The complete text is not included in the previous sample to eliminate repetition. When you try these applications for your own use, make sure you complete the entire text given in the typeset sample. Keep in mind that your page breaks may differ from the samples given depending on the typeface used. Remember—these are examples. You may discover an easier way to accomplish the same thing.

The job file for a two-column layout will appear similar to the one on the following page.

{**.LFD(bookfmt)**}{**start**}{**chaphead**}{**layout2**}<**FT9**>Information Overload.
<**FT7**>The contemporary business community finds itself beset by a superabundance of paper-based information. Myriad proposals, plans, memos, letters, contracts, reports, bulletins, and other printed communications cross each businessperson's desk each day, and this deluge is unlikely to recede in the foreseeable future.←
☐It has been estimated that more than 72 billion pieces of written business communication are being produced yearly. All of these documents, of course, need to be read, sorted, distributed, output, stored, and/or retrieved. Computerized office automation, which was heralded as the cure for business's communication headaches, is in fact creating <**FT8**>more <**FT7**>paper. That today's office is far from paperless is more readily evidenced by the explosive growth of overnight document delivery services such as Airborne and Federal Express. Similarly, corporate purchasing departments report that they are buying more printer and photocopier paper today than ever before.←
☐It has become evident that office communication simply cannot be productive in totally electronic form. A computer screen lacks the special "feel" and profitability of paper-based materials. Increasingly, businesspeople are finding that video display terminals are often impractical vehicles for everyday office communications.←
☐Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data::←
{**tab**}ᵀʙ1)ᵀʙemployees are unable to process all the information they receive
ᵀʙ2)ᵀʙemployees tend to disregard some information altogether
ᵀʙ3)ᵀʙemployees often divert information to inappropriate receivers
ᵀʙ4)ᵀʙemployees arbitrarily deal with information without regard to its relative importance

(Text continues on next page)

5) employees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences

6) employees attempt to "pigeonhole" information according to their own arbitrary categories

<**FT9**>The Typographic Solution.<**FT7**>Up to now, the technological advances responsible for automating office work. . . . .

The job file for a three-column layout will appear similar to that on this and the following page.

{**.LFD(bookfmt)**}{**start**}{**chaphead**}{**layout3**}<**FT9**>Information Overload.
<**FT7**>The contemporary business community finds itself beset by a superabundance of paper-based information. Myriad proposals, plans, memos, letters, contracts, reports, bulletins, and other printed communications cross each businessperson's desk each day, and this deluge is unlikely to recede in the foreseeable future.←
☐It has been estimated that more than 72 billion pieces of written business communication are being produced yearly. All of these documents, of course, need to be read, sorted, distributed, output, stored, and/or retrieved. Computerized office automation, which was heralded as the cure for business's communication headaches, is in fact creating <**FT8**>more <**FT7**>paper. That today's office is far from paperless is more readily evidenced by the explosive growth of overnight document delivery services such as Airborne and Federal Express. Similarly, corporate purchasing departments report that they are buying more printer and photocopier paper today than ever before.←
☐It has become evident that office communication simply cannot be productive in totally electronic form. A computer screen lacks the special "feel" and profitability of paper-based materials. Increasingly, businesspeople are finding that video display terminals are often impractical vehicles for everyday office communications.←

(Text continues on next page)

□Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data:←↵
{**tab**}T_B1)TBemployees are unable to process all the information they receive↵
T_B2)TBemployees tend to disregard some information altogether ↵
T_B3)TBemployees often divert information to inappropriate receivers↵
T_B4)TBemployees arbitrarily deal with information without regard to its relative importance↵
T_B5)TBemployees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences↵
T_B6)TBemployees attempt to "pigeonhole" information according to their own arbitrary categories↵
<**FT9**>The Typographic Solution.<**FT7**>Up to now, the technological advances responsible for automating office work. . . . .

☐Research studies indicate that our present excess of paper-based information frequently results in a serious breakdown of communication. Productivity experts have found that information overload can have a number of adverse consequences in the office environment. They have discovered, for example, that when confronted with an overabundance of incoming data:←↵
{**tab**}ᵀʙ1)ᵀʙemployees are unable to process all the information they receive↵
ᵀʙ2)ᵀʙemployees tend to disregard some information altogether ↵
ᵀʙ3)ᵀʙemployees often divert information to inappropriate receivers↵
ᵀʙ4)ᵀʙemployees arbitrarily deal with information without regard to its relative importance↵
ᵀʙ5)ᵀʙemployees inappropriately resort to standard, "off-the-shelf" responses without considering their ultimate consequences↵
ᵀʙ6)ᵀʙemployees attempt to "pigeonhole" information according to their own arbitrary categories↵

<**FT9**>The Typographic Solution.<**FT7**>Up to now, the technological advances responsible for automating office work. . . . .

REFERENCE

This section is designed to help you find information quickly. Included are:

- Power Page Features
- Step-by-step instructions for entering Format commands
- Step-by-step instructions for entering Area Composition commands.
- Error messages.

Power Page Features

Power Page is a software option that can be ordered for any PowerView 10.

The maximum number of characters that can be used in a format is 255. The maximum number of characters in each format file is 2,000 characters.

The following Master Program Disks are included with Power Page and must be included when creating your User Program.

Disk Description	Disk Name
Background Compose Option MPD	bgcm-pv
Format Option Disk MPD	fmt-pv
Paging Option MPD	page-pv
Vertical Justification Option MPD	vj-pv

Area Composition Commands

Defining A Page <PAGEw,d,i>

The page command establishes page mode. It must be the first entry in a text file. Maximum horizontal and vertical values depend on the range of your typesetter.

To define a page:

TYPE CMO PAGEwidth,depth,interpage gap TYPE CMX

To define a page with top alignment, add a T to the end of the Page command. If no value is entered, the alignment will default to base alignment.

<PAGEw,d,i,T>

Defining A Compartment <CMPTx,y,w,d,h,v>

A compartment is an area on a page, used for positioning text or figures. The maximum width and depth of a compartment may not exceed the width and depth of the defined page.

To define a compartment:

TYPE CMO CMPTindent from left margin,depth from top of page,width,depth, horizontal quad,vertical quad TYPE CMX

X/Y Coordinates <XYi,d>

With X/Y coordinates, you may position text or rules within a page. It is not necessary to have a compartment defined, but a page must be defined.

To enter X/Y coordinates:

TYPE CMO XY(indent from left margin),(depth from top of page) TYPE CMX

Vertical Insert Space <VIS> or <VISx>

The Vertical Insert Space command defines the point where excess vertical space is inserted during justification. It may be used with any vertical quad command.

To enter a Vertical Insert Space:

TYPE CMD VIS TYPE CMX or TYPE CMD VISx TYPE CMX

The *x* represents a proportional value that can be used to increase the amount of inserted space at specific points. For example, <VIS2> will expand twice as much as <VIS> or <VIS1>.

Vertical Band Set <VBSmin,max>

The Vertical Band Set establishes minimum and maximum values for vertical spacebands.

To define vertical spaceband values:

TYPE CMD VBSminimum value,maximum expansion value TYPE CMX

Vertical Band <VB>

This command enters a vertical band value previously set with a vertical band set command.

To enter a VB command:

TYPE CMD VB TYPE CMX

Carding <CRDv>

The carding command enters the maximum amount of space, in points, that the PowerView 10 will distribute over all lines of text within a compartment. The maximum carding value you may enter depends on the maximum depth range of your typesetter.

To enter a carding command:

[TYPE CMO] CRDvalue [TYPE CMX] after the compartment command.

Format Commands

In this section, you will find instructions for entering each Power Page command. They are given in sequential order, as presented in the manual.

Defining a Format

To create a format in a format file, use the following steps.

1. Open a new file.
2. Put your PowerView 10 in the Edit Mode.
3. Define a format:

   [PREC LC] [TYPE CMO] .DFMT,formatname [PREC LC] g contents of format [PREC UC] g [PREC UC] [TYPE CMX]

4. Append the page and close the file.

To define a format with PS14, LS16, FT1, LL30:

[PREC LC] [TYPE CMO] .DFMT,one [PREC LC] g <PS14<LS16<FT1<LL30> [PREC UC] g [PREC UC] [TYPE CMX]

This will appear on your source window:

{.DFMT,one(<PS14<LS16<FT1<LL30>)}

Verifying the Accuracy of a Format

After defining a format in your format file, you may want to verify the accuracy of that format.

1. Make sure your PowerView 10 is in the Edit Mode.
2. Check the formats:
   [FILE FUNC] CHKFD [EXECUTE]

Loading a Format File

1. Open a text file.
2. With the cursor at the beginning of the file, load the format file:
   [PREC LC] [TYPE CMD] .LFD [PREC LC] g name of format file [PREC UC] g [PREC UC] [TYPE CMX]
3. Type copy.
4. Append the page and close the file.

To load the format file named "format":
[PREC LC] [TYPE CMD] .LFD [PREC LC] g format [PREC UC] g [PREC UC] [TYPE CMX]

This will appear in your source window:

{.LFD(format)}

Calling a Format

1. Make sure the format file is loaded into the text file.
2. Call the format into the text file *after* the .LFD command:
   [PREC LC] [TYPE CMD] format name [PREC UC] [TYPE CMX]
3. Enter copy, if necessary, or replace the page and close the file.

For example, the format "one" is defined in format file "format". To call the format "one" into the text file:

1. Make sure the format file "format" is loaded.
2. After the .LFD command:
   [PREC LC] [TYPE CMO] one [PREC UC] [TYPE CMX]

The .LFD and format call will look like this on your source window:

{.LFD(formats)}{one}

Exploding a Format

After calling a format, the contents of the format may be "exploded" onto the screen. This command is useful for checking errors without recalling the format file. The PowerView 10 must be in Compose Mode.

1. Move the cursor to the closing brace (}) of a format call.
2. Explode the format:
   [FILE FUNC] EXFMT [EXECUTE]

Using Variables in a Format

1. Open a format file.
2. Define a format using the [DATA] key to show variables within the format:
   [PREC LC] [TYPE CMO] .DFMT,format name,, [PREC LC] g<PS [DATA] 1<LS [DATA] 2<FT1>
   <LL20> [PREC UC] g [PREC UC] [TYPE CMX]
3. Append the page and close the file.
4. Call the text file in which the format will be used.
5. Load the format file.
6. Call the format into the file entering the values for the variables after the format name. Separate the values with commas:
   [PREC LC] [TYPE CMO] format name,variable,variable [PREC UC] [TYPE CMX]
7. Replace the page and close the file.

For example:

1. Define a format called "fmt". Enter point size and linespace as variables.
   PREC LC TYPE CMO .DFMT,fmt,, PREC LC g<PS DATA 1<LS DATA 2<FT1<LL20>
   PREC UC g PREC UC TYPE CMX

This will appear on your source window:

{.DFMT,fmt,,(<PSⒹ 1<LSⒹ 2<FT1<LL20>)}

2. Append the page and close the format file.
3. Call the text file in which the format will be used.
4. Load the format file.
5. Call the format "fmt" into the file, using variables 10 and 12 for PS and LS, respectively:

   PREC LC TYPE CMO fmt,10,12 PREC UC TYPE CMX

Whatever is inserted after the first comma in the default area becomes the contents of Ⓓ*1. Data following the second comma becomes the contents of* Ⓓ*2, and so on.*

Escaping characters

You may wish to use the colon (:) as a punctuation mark in a format definition, or the colon and/or comma as punctuation marks in the variable defaults. These characters must be preceded with another colon or they will not print. If you are unsure whether or not to escape a particular character, precede it with a colon.

Suppose you want to use a colon as the punctuation in a salutation that is part of a format definition. The colon following the name is preceded by another colon, for example, "Dear John Smith::".

Nesting a Format

A call to a format may be contained within another format; this is called "nesting". A nested format is defined the same way you define any other format, using the .DFMT command.

1. Nest a format by calling it within another format definition:
   PREC LC TYPE CMO format name PREC UC TYPE CMX

Entering Counters

Counters automatically count the number of occurrences of certain functions. Counters may be entered within a format file or anywhere in a text file.

Enter a counter using one of these arithmetic functions:

<EQUC1,value>	Equal
<ADDC1,value>	Add
<SUBC1,value>	Subtract
<MULC1,value>	Multiply
<DIVC1,value>	Divide
<MODC1,value>	Supply a remainder when dividing

Dumping Counters

A counter will not display in a job unless it is "dumped". This is accomplished by entering the counter number in the .DMP command. The counter designation (C#) in a .DMP command must always be an uppercase C.

1. Place the cursor in the text where you want the counter to display.
2. Enter the .DMP command:
   PREC LC TYPE CMO .DMP,C# PREC UC TYPE CMX

Qualifying Counters

When a counter is dumped, it will display as an arabic number unless a counter qualifier is used in the .DMP command. The qualifier is entered after the counter designation. Qualify a counter using one of these functions:

Function	Qualifier
Suppress leading zeroes	S (default)
Output roman numerals	Upper (R) and lower case (r)
Output leading zeroes	(L#) Total digits including counter value (maximum: 9)
Output leading EN spaces	(N#) Total digits including counter value (maximum: 9)

Upper and lower case roman numerals may not exceed a value of 3,999. All counters will output as arabic numbers unless the qualifiers R or r are used. If no qualifier is entered, the MCS will use the default (S).

For example, to output a number with an upper case roman numeral qualifier, enter the following keystrokes:

PREC LC TYPE CMO .DMP,C#,R PREC UC TYPE CMX .

Using The Dump Command To Output A Value

Enter the dump command in a text file to output a value not defined in a counter. To output the year 1492 in roman numerals, your command will look like this:

{.DMP,1492,R}

Your output will look like this: MXDII

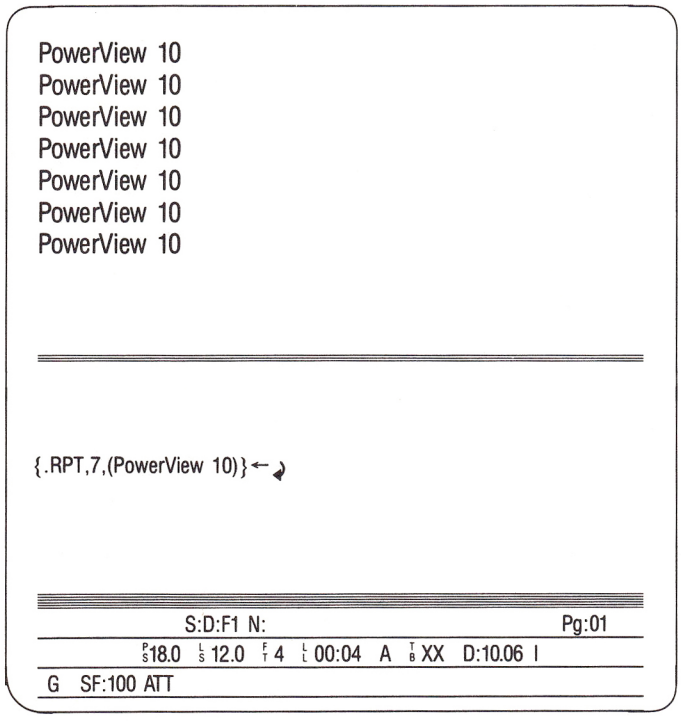
Repeat Function

The Repeat Function allows you to enter a string of characters once. The PowerView 10 will repeat the string as many times as you indicate in the .RPT command. The Repeat command may be entered in text or in a format file.

To enter the repeat command:

PREC LC TYPE CMO .RPT,#of times to repeat PREC LC g repeat string PREC UC g PREC UC TYPE CMX

If you are repeating the phrase POWERVIEW 10 seven times, the PowerView 10 screen will look like this:



You may use a previously-defined counter to indicate the number of times the repeat string will repeat.

PREC LC | TYPE CMO .RPT,C# PREC LC g repeat string PREC UC g PREC UC | TYPE CMX

The repeat string will look like this:

{.RPT,C#(repeat string)}

Compartment Break

A new compartment can be entered manually or automatically.

1. **Force Break** <**FB**> Copy will flow into next compartment or next page:
   TYPE CMO FB TYPE CMX

2. **New Page** <**NP**> Forces the start of a new page:
   TYPE CMO NP TYPE CMX

3. <?FB> System-entered compartment break appears in text automatically when breaking between linked compartments.

4. <?NP> System-entered page break appears in text automatically when breaking between pages.

REMEMBER, there can only be one output page per file page. When a new page command is entered either manually or automatically, or when the last compartment on the page is filled, the PowerView 10 will beep and you must begin a new file page.

Defining a Head Format <**HEAD::name::**> or <**HEAD:name:**>

To call a heading into a file, the heading must be defined in a head format.

1. TYPE CMO HEAD::name:: TYPE CMX. An escape character (colon) is used if the header is defined in a format file.

2. TYPE CMO HEAD:name: TYPE CMX. No escape character is needed if the header is defined in a text file.

Defining a Foot Format <FOOT::name::> or <FOOT:name:>

To call a footer into a file, the foot must be defined in a foot format.

1. [TYPE CMO] FOOT::name:: [TYPE CMX]. An escape character (colon) is used if the footer is defined in a format file.
2. [TYPE CMO] FOOT:name: [TYPE CMX]. No escape character is needed if the footer is defined in a text file.

Saving Parameters <SAVE>

Certain parameters may be saved to be used again in a different location. The SAVE command is entered following the parameter you wish to save for later use. Only one saved parameter can be active at a time.

The following parameters can be saved:

- Point Size
- Set Size
- Linespace
- Font I.D.
- Tab I.D.
- Pseudo Italic On
- Pseudo Italic Off

To save previous parameters:

[TYPE CMO] SAVE [TYPE CMX]

Restoring Parameters <RSTR>

The parameters saved with the SAVE command can be reactivated at a different location using the RSTR command.

To restore saved parameters:

[TYPE CMO] RSTR [TYPE CMX]

Messages

No Format	Error in format definition syntax. Appears when you try to call format into your text file. Open the format file, make necessary corrections and replace the file.
Invalid Command	Must be in Edit Mode to verify accuracy of formats. Must be in Compose Mode to explode a format.
Fmt Loaded	You have tried to load more than one format file.
Fmt Too Big	Format string contains more than 255 characters.
Fmt Syntax	Error made when entering .LFD command in file. When using CHKFD, syntax of indicated format contains an error(s). Error made when defining a Repeat or a Dump command. Trying to call a format that contains an error(s).
No File	Format file cannot be found. Must be present on same disk as text file or on system disk (F1 or H1).

Cntr Id	Counter number must be 1-9.
Cntr Value	Counter operation exceeds maximum value of 32,767.
Dup Format Name	Format file being loaded contains two formats with the same name.
Fmt Active	Format exploding (in the background) into the text stream when an Automatic New Page is generated by the PowerView 10. The exploding format must be entered so it will finish before or start after the page break.
Value Too Big	Parameter value specified in a counter operation exceeds 32,767.
FT??Error **PS??Error**	Parameter errors in format.
Nest Error	More than eight nesting levels used.
Galley Mode	Page command not the first item in the job. Move Page command to the beginning.
Page Window	Page definition exceeds maximum vertical or horizontal range of typesetter.
Page Exceeded	Defined compartment exceeds one or more of the page boundaries.

Page Overset	You have reached to end of the page and the PowerView 10 is automatically breaking the page for you. You must append or replace the current file page and begin a new file page.
End of Page	You have entered a manual page break. You must append or replace the current file page and begin a new file page.
Cmpt Limit	You have defined more than ten linked compartments.
Memory Full	2,000-character limit exceeded in format definition.
No Cmpt	No defined compartment.
Cmpt Exceeded	Specified X/Y position outside boundary of the compartment.
Page Open	Page dimensions previously defined.
VJO Cutrun	Vertical Justification must be turned off to define a Cut-runaround.

QUICK REFERENCE

FUNCTION	COMMAND PARAMETERS	EXPLANATION OF CODES
Auto Compartment Break	<?FB>	Automatically inserted by the system when breaking between linked compartments.
Automatic New Page	<?NP>	Automatically inserted by the system when breaking between pages.
Define Compartment	<CMPTx,y,w,d,h,v>	Defines the placement and dimensions of a compartment within a page. x = *horizontal coordinate (picas.points, or inches with I)* y = *vertical coordinate (picas.points, or inches with I)* w = *width (picas.points, or inches with I)* d = *depth (picas.points, or inches with I)* h = *horizontal quad mode* v = *vertical justification quad mode*
Define Page	<PAGEw,d,g,a>	Defines dimensions for *all* pages within a file. Must be entered at the beginning of the file, prior to entering any text or typesetting parameters. w = *width (picas.points, or inches with I)* d = *depth (picas.points, or inches with I)* g = *interpage gap (picas.points, or inches with I)* a = *alignment; T = top-alignment,* B = *base-alignment.*
Force Break	<FB>	Causes the current line and compartment end and the next compartment to become active.
New Page	<NP>	Causes an immediate "page eject" to occur. Ends the current page and begins a new page.
Page Castoff	FILE FUNC *page* EXECUTE	Breaks the file pages into "output size" pages.
Background Compose	QUEUE *bg*	Background function. Breaks file into file pages that match the output pages. Display BG can be used to monitor the process.
Restore	<RSTR>	Reactivates previously "saved" typesetting parameters.

FUNCTION	COMMAND PARAMETERS	EXPLANATION OF CODES
Save	<SAVE>	Parameter values "saved" are point size, set size, line space, font id., tab id., pseudo italic on/off and pseudo italic set. Stores the values defined for the preceding parameters in the "save area" of the system. Only one "set" of parameters may be saved at any given time.
Specify Footer Format	<FOOT:format name:>	Specifies format to be used at all footer points. Footer format *must* exist in loaded format file. If footer is specified in a format definition, two colons *must* be used for delimiters (one as the escape character).
Specify Header Format	<HEAD:format name:>	Specifies format to be used at all header points. Header format *must* exist in loaded format file. If header is specified in a format definition, two colons *must* be used for delimiters (one as the escape character).
VJ: Carding	<CRD#>	Defines the maximum amount of Carding space. # = *maximum value (points.quarter points).*
VJ: Vertical Band	<VB>	Places a Vertical Band (value defined in Vertical Band Set) in the copy at the point of insertion.
VJ: Vertical Band Set	<VBSn,m>	Defines the minimum and maximum expansion values used when a <VB> is entered. n = *minimum value (points.quarter points)* m = *expansion range value (points.quarter points).*
VJ: Vertical Insert Space	<VIS> or <VISx>	Defines the point(s) where excess vertical space will be distributed. x *(optional) = a proportional value; used to increase the amount of excess space at specific points.*
X/Y Position	<XYx,y>	Used to define where copy will be positioned within a page or compartment. Based on left/upper margin of page/compartment. x = *horizontal coordinate (picas.points, or inches with I)* y = *vertical coordinate (picas.points, or inches with I).*

INDEX OF SUBJECTS

A

About this Manual I–1
Area Composition III–1
Alternating Page Layouts V–16-25
Applications Section VI

B

Business Cards Applications VI–16-19

C

Calling a Format IV–12-14; VII–5
Carding III–31-33; VII–4
Check Format Definition Syntax IV–8; VII–5
Compartment III–3, 12-19; VII–2
  Defaults III–17
  Force Break Command III–18; VII–12
  Linking III–14
  Screen Messages VII–19-21
Complex Documents V–27
Counters IV–35-42; VII–9-10
Creating a Format File IV–3-7

D

Defining a Compartment III–12-14
Defining a Format IV–4; VII–4
Defining a Page III–6, 7; VII–2
Defining a Top Aligned Page III–9-11
Display Ad Applications VI–11-12, 15–16
Dump Command (Counters) IV–36-40; VII–9-10

E

Error Messages VII–19-21
Escape Characters IV–23; VII–8
Exploding a Format IV–14; VII–6

F

File Management V–27
Foot Command V–4; VII–13
Force Break Command III–18; VII–12
Format Overview IV–1
  How Are Formats Created And Used? IV–2
  How Does it Differ From A UDK? IV–1
  What Is A Format? IV–1
  Why Is It Used? IV–1
Format
  Calling IV–12-14; VII–5
  Definition IV–4; VII–4
  Escaping IV–23; VII–8
  Exploding IV–14; VII–6
  Files IV–3, 5-7; VII–4-5
  Limitations IV–5; VII–1
  Loading IV–9; VII–5
  Nesting IV–28-31; VII–9
  Nesting and Variables IV–31-35
  Option Disks VII–1
  Screen Messages VII–19-21
  Syntax Check Procedure IV–8; VII–5
  Variables IV–19-22, 32-35; VII–7-8
  Variables with Defaults IV–26

G

Getting Started II–1

H

Head Command V–5; VII–12
How to Use this Manual I–1

I

Inch Counting II–2

L

Linking Compartments III–14-16
Loading a Format File IV–9; VII–5

M

Magazine Page Layout Applications VI–19-21
Master Program Disks VII–1
Memory Requirements VII–1
Messages (Screen) VII–19-21
Metric Counting II–2
Multiple Page Documents V
  Applications VI–22-36
  Foot Command V–4; VII–13
  Head Command V–5; VII–12
  Screen Messages VII–19-21

N

Nesting Formats IV–28-31; VII–9
Numbering (Counters) IV–35-42; VII–9-10

O

Outline Applications VI–5-6

P

Page III–2, 6
Page Numbering IV–35-42
Page Numbers That Do Not Start at 1 . . . . . V–26
Pagination V–1
Paging Option MPD VII–1
Passing Variables to Nested Formats IV–32-35
Price List Applications VI–9-10
Power Page Features VII–1
Publications Applications VI–1-4

Q

Qualifiers in Commands II–2
Qualifying Counter Output IV–41-42

R
Repeat Command IV–43-44; VII–10-11
Restore Parameters Command V–15; VII–13
Ruled Form Applications VI–13-14

S
Save Parameters Command V–14; VII–13
Schedule Applications VI–7-8
Screen Memory VII–1
Screen Messages VII–19-21
Syntax, Format IV–11; VII–5

T
Top Align III–9-10

U
User Program Setup II–1; VII–1

V
Vertical Bands III–26-29; VII–3
Vertical and Horizontal Justification
Comparison III–23-24
Vertical Insert Space III–25; VII–3
Vertical Justification III–5, 23-32
Vertical Justification Carding III–30-32; VII–4
Vertical Justification MPD VII–1
Vertical Justification On/Off III–25

W
What's in the Manual I–2

X
XY Coordinates III–2-4, 20-22; VII–2